POPULATION GENETIC INFERENCE WHEN MUTATION RATES ARE CONTEXT-DEPENDENT

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Ryan Daniel Hernandez

January 2008

POPULATION GENETIC INFERENCE WHEN MUTATION RATES ARE CONTEXT-DEPENDENT

Ryan Daniel Hernandez, Ph.D.

Cornell University 2008

Population genetic studies often analyze patterns of single nucleotide polymorphisms (SNPs) to gain insight into the evolutionary history of a population. One summary statistic that has proved invaluable in these efforts is the frequency distribution of derived mutations (*i.e.*, the site-frequency spectrum, or SFS). In order to generate the SFS, orthologous sequences from closely related outgroup species are frequently used to distinguish ancestral and derived alleles at each SNP (assuming the ancestral allele is the one that matches the outgroup).

In a series of studies, I test the robustness of the parsimony assumption to a more realistic finite-sites model of context-dependent mutation biases inferred along the human lineage. I show (using both simulations and a theoretical model) that enough unobserved substitutions could have occurred since the divergence of human and chimpanzee to cause a shift in the SFS. The shifted SFS induced by misidentifying the ancestral states of some SNPs can lead to poor fitting demographic models and cause many statistical tests to spuriously reject neutrality in favor of models with positive selection.

By constructing a novel model of the context-dependent mutation process, polymorphism data can be corrected for the effect of ancestral misidentification. Using this correction, statistical tests return to their proper rejection rates, allowing for more accurate inference of both demographic events as well as the strength and abundance of natural selection. This correction is used to better understand

the evolution of GC-content in the human genome, and to perform accurate demographic inference in two populations of the biomedically important rhesus macaque.

Finally, I present a new forward simulation program, `SFS_CODE`, that can simulate several populations under a Wright-Fisher style island model. This program is highly flexible, allowing the user to simulate several loci (with or without linkage), where each locus can be annotated as either coding or non-coding, sex or autosome, selected or neutral. In addition to providing the source code for our program, we have also developed a web server that will allow the user to perform simulations using the high performance computing resources of the Computational Biology Service Unit at Cornell University (`http://cbsuapps.tc.cornell.edu/sfscode.aspx`).

BIOGRAPHICAL SKETCH

Ryan Daniel Hernandez was born on February 14, 1981 to parents Mary Linda and Robert Dennis Hernandez. He grew up in San Leandro, CA with his two siblings, Jeannine and Russell. Approximately twelve houses down the road lived his late paternal grandparents, Esther and Paul, whom he visited almost daily throughout his youth.

Unlike many who would seek advanced scientific degrees, Ryan grew up much more interested in thinking about sports than pondering the nature of things. In high school, Ryan was a dedicated wrestler and football player, helping to lead both teams to league championships (due in no small part to *Jackson, CA*). It was not until his twelfth grade pre-calculus teacher told him that a degree in mathematics would make him "hirable in any field" that he was really convinced to be the first person in his family to move away from home to go to college. In 1999, Ryan graduated from San Leandro High School, missing the top ten percentile by a single individual. Ryan then enrolled at Pitzer College (a member of the Claremont Colleges), where he spent two years in a nearly schizophrenic state, bouncing back and forth between being a football player and a would-be academic.

In his first semester at Pitzer College, Ryan was motivated by an outstanding calculus professor, Dr. Judith Grabiner. It was in her class that he began to ask *why* certain formulas were true and why they worked. Needless to say, this got Ryan started toward a degree in mathematics. The very same semester, Ryan came across an advertisement for the Mathematical and Theoretical Biology Institute (MTBI) at Cornell University. The ad claimed to pay undergraduates of minority descent to play with math for a summer. Intrigued, Ryan kept the card for an entire year until he was eligible to apply.

MTBI is organized by Dr. Carlos Castillo-Chavez (C^3), and consists of an intensive month of coursework followed by a three-week research project. The experience changed Ryan's life. It was during this summer program that Ryan realized that he was not only capable of doing mathematics at a high level, but also that he should go to graduate school (and be paid to do so!). Now driven by a strong desire to focus more on developing his brain than his brawn, he gave up his position as the starting wide receiver on the Pomona-Pitzer Sagehens football team (they subsequently lost more than half of their games for the first time in nearly a decade).

Ryan went on to receive honors recognition for his B.A. in mathematics in 2003. Now set on a life of math and its applications, Ryan applied to the Center for Applied Mathematics graduate program at Cornell University. However, due to a somewhat mysterious sleight of hand, Ryan received an acceptance letter from the graduate field in Biometry at Cornell, where he spent the next four and half years studying population genetics under the direction of Dr. Carlos Bustamante.

ACKNOWLEDGMENTS

It is with great honor that I thank a few of the people that have contributed to my development, both as a person and as a scientist.

First off, this dissertation would not have been possible if it were not for my adviser and mentor, Carlos Bustamante. Over the last four and a half years, Carlos has provided an unparalleled amount of creative energy. His eagerness to start new projects has helped me to develop the necessary skills of simultaneously managing many large-scale studies, and by pushing me to study the population genetics of humans, rhesus macaques, dogs, cows, and even rice, he has instilled in me a broad understanding of patterns of genetic variation.

I would also like to thank my committee members, Andy Clark and Rick Durrett, as well as the many current and former members of the Bustamante lab (now too many to list) for helpful discussions and encouragement. Without your support, my time at Cornell would have been quite rough. I would especially like to thank Scott Williamson (a postdoc in the lab when I started, and a professor in the department when I graduated), whose creative modeling techniques and insight helped to kick start my research career. Thanks must also be sent to Rasmus Nielsen, whose insightful comments never ceased to amaze me.

It is also important to acknowledge my family, who helped me to get by when times were rough. Though calling to say "hello" is generally not my best quality, you always provide a happy, comforting voice when I do. In addition, I must thank all of my friends at Cornell who have helped to make sure that I was not wasting away in front of my computer all day and night. Finally, a special heartfelt thanks must also go to Dara Torgerson, my partner through the last half of my PhD. Dara opened my eyes to a wonderful life of fine food and wine, and helped me over each and every hurdle I faced along the way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

CONTEXT DEPENDENCE, ANCESTRAL MISIDENTIFICATION, AND SPURIOUS SIGNATURES OF NATURAL SELECTION*

*Originally published as: Hernandez, R. D., S. H. Williamson, and C. D. Bustamante (2007). Context dependence, ancestral misidentification, and spurious signatures of natural selection. *Mol Biol Evol*, 24(8):1792–1800, doi: 10.1093/molbev/msm108.

1.1 Abstract

Population genetic analyses often use polymorphism data from one species, and orthologous genomic sequences from closely related outgroup species. These outgroup sequences are frequently used to identify ancestral alleles at segregating sites and to compare the patterns of polymorphism and divergence. Inherent in such studies is the assumption of parsimony, which posits that the ancestral state of each single nucleotide polymorphism (SNP) is the allele that matches the orthologous site in the outgroup sequence, and that all nucleotide substitutions between species have been observed. This study tests the effect of violating the parsimony assumption when mutation rates vary across sites and over time. Using a context-dependent mutation model that accounts for elevated mutation rates at CpG dinucleotides, increased propensity for transitional versus transversional mutations, as well as other directional and contextual mutation biases estimated along the human lineage, we show (using both simulations and a theoretical model) that enough unobserved substitutions could have occurred since the divergence of human and chimpanzee to cause many statistical tests to spuriously reject neutrality. Moreover, using both the chimpanzee and rhesus macaque genomes to parsimoniously identify ancestral states causes a large fraction of the data to be removed while not completely alleviating problem. By constructing a novel model of the context-dependent mutation process, we can correct polymorphism data for the effect of ancestral misidentification using a single outgroup.

1.2 Introduction

Identifying the action of natural selection from patterns of standing genetic variation has long been of interest to the population genetic community. The recent

confluence of genomic data and the requisite computational power for large scale analyses now make this important goal a tractable problem. Of the many statistical methods developed to detect the presence and infer the strength of natural selection, those that make use of the frequency distribution of all derived mutations observed in a sample of chromosomes (the unfolded site-frequency spectrum, or SFS) appear to be the most powerful (Sawyer and Hartl, 1992; Hartl et al., 1994; Akashi, 1999; Bustamante et al., 2001; Nielsen et al., 2005b).

One technique that has been applied to make analysis of the SFS more robust to demographic and non-stationary evolutionary processes is to compare the SFS in a region of interest to the SFS in a genomic region putatively untouched by natural selection (Akashi, 1999; Williamson et al., 2005). However, as large scale human polymorphism data has become available, a striking pattern has emerged. Figure 1.1 shows the normalized SFS expected under the standard neutral model [SNM; Watterson (1975); Hudson (1990); Fu (1995)] and the normalized SFS for the observed non-coding and synonymous mutations from 161 gene regions sequenced across a world-wide panel of 95 humans [NIEHS panel 2, as described in Hernandez et al. (2007c)]. A visual inspection of Figure 1.1 suggests that the observed data show an overrepresentation of both low and high frequency mutant alleles and a paucity of intermediate ones compared to the SNM.

An observed skew toward rare alleles is generally attributed to either the effect of natural selection restricting the spread of slightly deleterious mutations (Fu and Li, 1993; Williamson and Orive, 2002), or to the effect of a growing population where most mutations tend to be young (Slatkin and Hudson, 1991; Griffiths and Tavaré, 1994). A relative excess of mutant alleles at high frequency compared to the SNM is typically attributed to the presence of recurrent positive natural selection (Akashi and Schaeffer, 1997; Hartl et al., 1994; Bustamante et al., 2001),

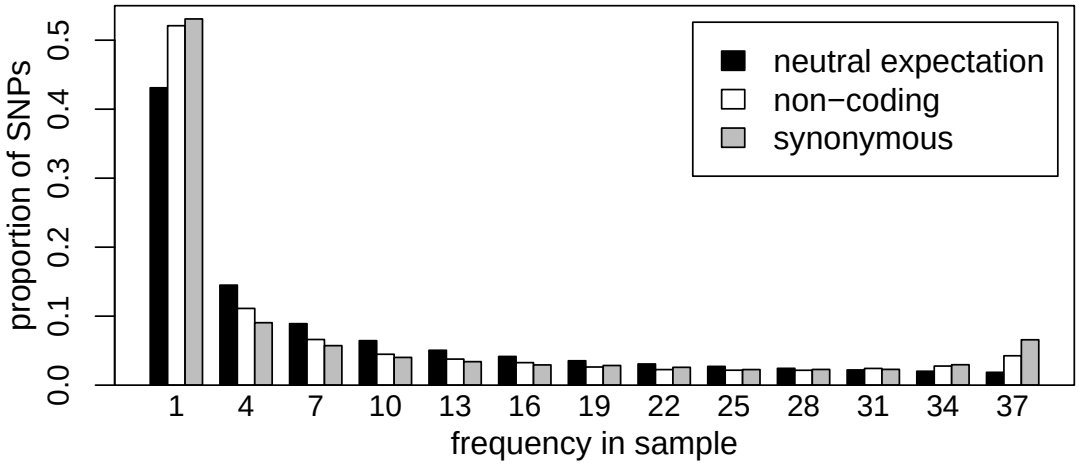

Figure 1.1: The normalized SFS expected under the standard neutral model and observed in putatively neutral regions (non-coding and synonymous). SNPs with missing data were incorporated by generating the expected SFS in a sub sample of 40 chromosomes Marth et al. (2004); Nielsen et al. (2004), then all SNPs were pooled into frequency bins of size three.

linkage to the target of a recent selective sweep (Kim and Stephan, 2000; Fay and Wu, 2000), or the existence of hidden population substructure with low levels of migration (Wakeley and Aliacar, 2001). Because the data used here includes several individuals from each population, hidden population structure is unlikely to have caused the excess high frequency tail, therefore an excess of both low and high frequency mutant alleles casts doubt upon the assumption of selective neutrality for both non-coding and synonymous mutations.

Another explanation for the relative excess of high frequency mutations in putatively neutral regions is based on violations of the parsimony assumption used to construct the SFS. Namely, if a substitution occurred during the divergence time of the two species being compared, and a subsequent polymorphism (back mutation) recently arose at the same site, then the nucleotide observed in the outgroup would not be the ancestral state of the SNP. Since most mutations are transitions, multiple mutations at a site (multiple hits) would often be back mutations, thus

ancestral misidentification would often times go unnoticed. Because most derived mutations are expected to be rare (Watterson, 1975; Fu, 1995), ancestral misidentification would most often result in mislabeling low frequency mutations as high frequency ones (Baudry and Depaulis, 2003).

Previous studies that have addressed multiple hits have either focused on correctly counting the number of substitutions between two diverged sequences (Jukes and Cantor, 1969; Kimura, 1980), or correctly counting the number of segregating mutations in a sample of sequences from hypervariable regions (Tajima, 1996; Yang, 1996). These studies typically assume that all sites are evolving homogeneously and at stationarity, but may allow for mutation rate variation across sites (though constant over time). However, recent analysis of mammalian sequences have revealed mutational patterns that are much more complex than these models account for. In particular, the mutation rate at a site appears to depend on the source and target nucleotides, as well as the adjacent nucleotides 5' and 3' of the site (Hwang and Green, 2004; Siepel and Haussler, 2004). Under such a context-dependent model, site-specific mutation rates can vary in magnitude as much as 50-fold (Blake et al., 1992; Hess et al., 1994; Hwang and Green, 2004; Siepel and Haussler, 2004). The largest change in site-specific mutation rates is due to the production (or removal) of a methylated CpG dinucleotide (the so called "CpG-effect"). Surprisingly, non-CpG sites can also vary as much four-fold depending on their context (Hwang and Green, 2004). By taking the complexities of such a mutation process into account, we derive a mathematical model for the SFS that can correct for ancestral misidentification.

1.3 Materials and Methods

1.3.1 Theory

In this study, we consider a context-dependent nucleotide mutation model. This model assumes that the rate of mutation at a nucleotide site depends on its flanking nucleotide context (*i.e.*, its 5' and 3' neighboring nucleotides), and is composed of sixteen 4×4 nucleotide mutation rate matrices (one for each pair of flanking nucleotides). For ease of notation, we consider $\mathcal{Q}$, the 64×64 instantaneous trinucleotide mutation rate matrix that is restricted to only allow changes at the second position of each trinucleotide. Hwang and Green (2004) estimated the parameters of such a context-dependent mutation model assuming non-reversiblity and strand-symmetry from an untranscribed DNA sequence of length 5.2 Mb across 19 mammalian species. In our applications, we will use their estimates obtained along the human lineage, but our derivations will be sufficiently general to allow for any model of mutation rate variation (subject to additional assumptions as necessary).

The probability of substituting one nucleotide for another over a time interval t_s in a fixed context (*i.e.*, when both 5' and 3' nucleotides do not change), can be obtained from the probability substitution matrix $\mathcal{P}(t_s)$, where $\mathcal{P}(t_s) = \exp(\mathcal{Q}t_s) = \sum_{n=1}^{\infty}(Qt_s)^n/n!$. Note that time ($t_s$) is scaled in terms of the expected number of substitutions per site, and that the main diagonal of $\mathcal{Q}$ was set so as to satisfy the mathematical requirement that each row sums to zero.

Probability of Ancestry: We will derive the probability of correctly identifying the ancestral state of a SNP when the orthologous site in the outgroup matches one of the segregating alleles. Let M be the unknown ancestral state of the SNP, with S the two segregating alleles (an unordered pair) and U the allele observed in the outgroup. Assume that the divergence time between species (t_s) is

scaled in terms of the number of substitutions per site and known. For simplicity, and to accommodate the structure of our context-dependent mutation model, we assume that the nucleotides flanking the polymorphism have remained constant since the divergence of the two species being compared (henceforth referred to as the constant context assumption).

For the arbitrary case of observing an allele $U = u$ in the outgroup while the pair of alleles $S = \{u, x\}$ are segregating in the population, the probability of correctly identifying the ancestral state of the SNP, ν_{ux}, is

$$\begin{aligned} \nu_{ux} &= P(M = u \mid U = u, S = \{u, x\}, t_s) \\ &= \frac{P(M = u, U = u, S = \{u, x\} \mid t_s)}{P(M = u, U = u, S = \{u, x\} \mid t_s) + P(M = x, U = u, S = \{u, x\} \mid t_s)}. \end{aligned} \tag{1.1}$$

This follows from conditional probability under the assumption that either $M = u$ or $M = x$ (equivalent to an infinite-sites assumption for within species polymorphism). Each of the terms in equation (1.1) can be simplified by reordering terms and applying conditional probability to sequences at stationarity for trinucleotide frequencies. For example, in the numerator we have

$$\begin{aligned} & P(M = u, U = u, S = \{u, x\} \mid t_s) \\ & = P(U = u \mid t_s) P(M = u \mid U = u, t_s) P(S = \{u, x\} \mid M = u, U = u, t_s) \\ & = P(U = u) P(M = u \mid U = u, t_s) P(S = \{u, x\} \mid M = u) \\ & = \begin{cases} \pi_u \mathcal{P}_{uu}(t_s) \frac{\mathcal{Q}_{ux}}{\sum_{i \neq u} \mathcal{Q}_{ui}} & \text{if time-reversible} \\ \sum_\alpha \pi_\alpha \mathcal{P}^2_{\alpha u}\left(\frac{t_s}{2}\right) \frac{\mathcal{Q}_{ux}}{\sum_{i \neq u} \mathcal{Q}_{ui}} & \text{else} \end{cases} \end{aligned} \tag{1.2}$$

where the second equality follows from the assumption of stationary trinucleotide frequencies since the time of divergence (first term), and that the type of SNP is independent of any outgroup information when conditioning on its ancestral state (third term). Note that by conditioning on the ancestral state of a polymorphic site to be u, we can treat the possible derived states as competing exponential processes. This model has been well studied, and it is known that the probability

that the mutation will be of type $u \rightarrow x$ is just the ratio of its rate to the sum of all competing rates. In the final equality, time-reversibility refers to a model in which $\pi_u \mathcal{P}_{ux}(t_s) = \pi_x \mathcal{P}_{xu}(t_s)$ for all u and x, and that in the case of non-reversibility we sum over α, the four possible allelic states that could have occupied the most recent common ancestor (MRCA) of both species while maintaining the constant context assumption. Also note that in the non time-reversible case, we have assumed that the branches leading to both species are equal, but this need not be true and is easily relaxed by conditioning on the two branch lengths. The probability of correctly identifying the ancestral state of a SNP using outgroup information is then obtained by substituting equation (1.2) and a similarly derived term into equation (1.1).

Correcting the SFS: We consider the SFS constructed from a sample of n chromosomes with orthologous outgroup information. For all pairs of trinucleotides (generically denoted u and x) that meet the constant context assumption, let $N_{ux}(i)$ be the number of diallelic sites at which i chromosomes carry allele x, and the remaining $n - i$ chromosomes as well as the outgroup carry the allele u. $N_{ux}(\cdot)$ would then be the SFS for the inferred $u \rightarrow x$ mutations. However, because of ancestral misidentification, $N_{ux}(\cdot)$ is not necessarily the SFS for all SNPs of type $u \rightarrow x$. We therefore model $N_{ux}(i)$ as a mixture of the number of true $u \rightarrow x$ mutations at frequency i whose ancestral states were correctly identified [an unknown number denoted by $R_{ux}(i)$] and the number of true $x \rightarrow u$ mutations at frequency $n - i$ whose ancestral states were misidentified [another unknown number, denoted $R_{xu}(n - i)$].

The proportion of mutations from $u \rightarrow x$ whose ancestral states were correctly identified, f_{ux}, can then be written as the relative probability of observing u in the outgroup and a $u \rightarrow x$ mutation segregating in the population [similar to equation

(1.1)]. Namely,

$$f_{ux} = \frac{P(M=u, U=u, S=\{u,x\} \mid t_s)}{P(M=u, U=u, S=\{u,x\} \mid t_s) + P(M=u, U=x, S=\{u,x\} \mid t_s)}. \tag{1.3}$$

The number of $u \rightarrow x$ mutations at frequency i whose ancestral states were correctly identified is then $f_{ux}R_{ux}(i)$, and the number of $x \rightarrow u$ mutations at frequency $n-i$ whose ancestral states were misidentified is then $(1-f_{xu})R_{xu}(n-i)$. We can then write the observed SFS as a function of the true SFS as follows:

$$\begin{aligned} N_{ux}(i) &= f_{ux}R_{ux}(i) + (1-f_{xu})R_{xu}(n-i) \\ N_{xu}(n-i) &= f_{xu}R_{xu}(n-i) + (1-f_{ux})R_{ux}(i). \end{aligned} \tag{1.4}$$

This is a system of two equations with two unknowns [$R_{ux}(i)$ and $R_{xu}(n-i)$] that can readily be solved to give the reconstituted SFS as a function of the observed quantities

$$R_{ux}(i) = \frac{f_{xu}N_{ux}(i) - (1-f_{xu})N_{xu}(n-i)}{f_{ux}+f_{xu}-1} \tag{1.5}$$

$$R_{xu}(n-i) = \frac{f_{ux}N_{xu}(n-i) - (1-f_{ux})N_{ux}(i)}{f_{ux}+f_{xu}-1}. \tag{1.6}$$

To obtain the ith entry of the corrected SFS, $F_c(i)$, we then sum equations (1.5) and (1.6) over all pairs of trinucleotides u and x that meet the constant context assumption

$$F_c(i) = \sum_{\substack{(u,x): \\ u \neq x}} R_{ux}(i). \tag{1.7}$$

There are three steps to implementing this correction:

1. Tally the number of SNPs in each trinucleotide mutation class at each frequency [*e.g.*, $N_{\mathrm{ACG}\rightarrow\mathrm{ATG}}(i)$ = the number of ACG→ATG SNPs with i chromosomes carrying ATG and $N_{\mathrm{ATG}\rightarrow\mathrm{ACG}}(n-i)$ = the number of ATG→ACG SNPs with $n-i$ chromosomes carrying ACG]

2. Calculate the probabilities of ancestral misidentification from equation 1.3 (*e.g.*, $f_{\text{ACG}\to\text{ATG}}$ and $f_{\text{ATG}\to\text{ACG}}$)

3. Plug the results from step 1 and 2 into equations 1.5 and 1.6, and sum across all pairs of trinucleotides.

1.3.2 Simulations

To test the effect of a context-dependent mutation model with and without natural selection on our ability to compare closely related sequences, we simulated population genetic data with an outgroup. Our simulation program was written in the `C` programming language, and can be described as generating two Wright-Fisher populations with a known divergence time under a context-dependent mutation process across finitely many sites. The structure of the simulation is as follows: a single diploid population of constant size N_e is evolved for a large number of generations ($\geq 8N_e$) to mix and introduce variation, at which point there is a speciation event that splits the population into two independent populations of constant size N_e. After $2N_e\tau$ generations (the divergence time), a sample is taken from both populations. From these samples, several statistics are computed (*e.g.*, Tajima's D, Fu and Li's D, Fay and Wu's H, etc. as described below).

Each simulated generation consisted of 3 main components: (*i*) random mating, whereby the chromosomes of each diploid individual were chosen from two individuals of the previous generation (with replacement) with probabilities given by their relative fitness (assuming fitnesses are multiplicative across sites), (*ii*) mutation, whereby a Poisson number of events occurred with mean $\theta/2 = 2N_e\mu$ (the population scaled mutation rate per sequence per generation), and (*iii*) recombination, whereby a Poisson number of events occurred with mean $\rho/2 = 2N_e r$ (the population scaled recombination rate per sequence per generation).

Recombination events were distributed uniformly across individuals and sites (*i.e.*, no hot spots). Mutation events were distributed uniformly among all chromosomes in the population, but sites within chromosomes were chosen according to each site's "hit probability" (*i.e.*, the site's mutation rate relative to the rest of the sequence). Because the mutation rate at a site is context-dependent, a mutation event at a flanking nucleotide can change a site's hit probability as well as the overall mutability of a sequence (though only nominally for sufficiently long sequences). After choosing a site to mutate, a new nucleotide was chosen according to the relative mutation rate from the current state into each of the other three states (conditional on the flanking nucleotides). If the chosen mutation event resulted in an amino acid change and a selective effect was desired, then the derived state was also assigned a fitness coefficient from a specified distribution (*e.g.*, discrete, Gamma, Normal, or a more complicated mixture model). Should the mutation event result in the genesis of a stop codon, a different nucleotide was chosen for this site using the same probabilities as before. Although our algorithm for avoiding stop codons reduces the state space for some mutations, and slightly alters the resulting stationary trinucleotide frequencies, this does not induce a systematic bias in terms of polymorphism or divergence since a mutation is still introduced into the population.

In order to accommodate our assumption of statistical stationarity, we seeded our founding population with a single sequence drawn from the stationary distribution induced by the context-dependent mutation model. To obtain such a sequence, we first determined the stationary distribution of nucleotides for each context, π, which is the diagonal of $\lim_{t_s \to \infty} \exp(\mathcal{Q} t_s)$. By construction, each nucleotide context has four entries in π whose values correspond to the probability of each nucleotide in the given context. We then initialized each site of a sequence

to A, C, G, or T randomly, and then updated each site of the sequence by drawing a new nucleotide from π corresponding to the site's current nucleotide context. In order to avoid problems at the boundary of a sequence, we considered the sequence to be circular during the updating phase (*i.e.*, the first and last site were considered to be neighbors). We continued to update the sequence until trinucleotide frequencies converged to stationarity. Convergence to stationarity was monitored using the $\sqrt{\hat{R}}$ statistic (Gelman et al., 2004). Convergence to $\sqrt{\hat{R}} \in (.99, 1.01)$ for each of the 64 trinucleotide frequencies was ensured for 100 independent chains after 500 iterations (*i.e.*, $500L$ nucleotide updates).

Unless otherwise noted, all simulation results discussed were obtained assuming two diploid populations with effective size of $N_e = 250$, where $n = 50$ chromosomes were sampled from one population and $n = 2$ chromosomes were sampled from the outgroup (both without replacement). In agreement with diffusion theory (Sawyer and Hartl, 1992; Ewens, 2004), neutral coalescent theory (Kingman, 1982; Hudson, 1990), and previous forward simulation studies (McVean and Charlesworth, 2000; Williamson and Orive, 2002), we found that the actual population size does not impact our results when the mutation rate, recombination rate, divergence time, and selection coefficients are scaled by the effective population size (tested using $N_e = 250$, 500, 1000, and 5000, results not shown).

Implementation of our forward simulation was done by storing a single consensus sequence for each population, and two Splay trees [efficient self-balancing binary search trees; Sleator and Tarjan (1985)] for each individual. Each tree kept track of the mutations that an individual carried on a chromosome (including information regarding the generation in which they arose, whether or not they were synonymous, ancestral and derived codons, the states of the flanking nucleotides, the selective effect, as well as the true ancestral state). An individual inherited an

entire tree from each parent, and the consensus sequence was updated after each fixation event, at which point the reference to that mutation was removed from the trees of each individual and stored in a separate Splay tree. Such a simulation allowed us to evaluate the effect of ancestral misidentification directly, since the true ancestral state of each SNP was known, and we had outgroup information.

In order to evaluate the effect of natural selection, we implemented three "shift models" of selection. In a shift model, the selection coefficient at a site returns to zero when a mutation fixes in the population (*i.e.*, fitness is relative). In our simulations, the population genetic selection coefficient $\gamma = 2N_e s$ was drawn for each nonsynonymous mutation. The first selective scheme (γ_1) is a Normal-shift model where $\gamma_1 \sim N(0, 2)$. Normal-shift models are generally considered to represent positive selection, as a new mutation will increase the individual's fitness half of the time, and positively selected mutations are much more likely to be maintained/fixed in the population (Cutler, 2000). The second selective scheme (γ_2) refers to a gamma-shift model, where all nonsynonymous mutations are deleterious with $\gamma_2 \sim -\Gamma(1, .25)$ (this is a model of negative selection, where Γ is the common gamma distribution with mean 4, but reflected across the y-axis). The third selection scheme (γ_3) refers to a mixture of normals model with a point mass at zero, where most mutations are neutral (54.5% of nonsynonymous mutations have $\gamma_3 = 0$) or strongly deleterious [45% of nonsynonymous mutations have $\gamma_3 \sim N(-10, 5)$], but a small fraction are strongly advantageous [0.5% of nonsynonymous mutations have $\gamma_3 \sim N(50, 5)$]. Note that in each of these models, synonymous sites are completely neutral (*i.e.*, $\gamma_{syn} = 0$).

In order to contrast our simulations under the context-dependent mutation model with a nucleotide mutation model having no context effects, we simulated evolution under a generalized Kimura two-parameter model (Kimura, 1980). In

this model, each nucleotide has its own transition and transversion mutation rates, but there were no context effects such as the hypermutability of CpG dinucleotides. The parameters of this model were estimated by Zhang and Gerstein (2003) from 1,726 human ribosomal protein pseudogenes.

We verified the neutral component of our simulation study by comparing the number of segregating sites, fixed differences, and the SFS to those generated under the coalescent simulation program `ms` (Hudson, 2002). In Table 1.1, we show that the distribution of the observed number of segregating sites and fixed differences (with short divergence times) in our simulation matched the expected distribution very closely. However, as the divergence time increases, the number of unobserved substitutions grows substantially. Because we store information regarding all mutations during the simulation, we can see that the total number of substitutions that occurred matches the number we expect to see very closely, but due to multiple substitutions per site, the observed number of fixed differences is much less than expected.

Simulations for Poisson Random Field Analysis: A statistical test for positive selection using the SFS under the Poisson random field (PRF) framework of Sawyer and Hartl (1992) was recently proposed (Nielsen et al., 2005a). This test assumes that each nonlethal mutation enters the population and is assigned to one of three categories: neutral (with population scaled selection coefficient $\gamma = 2N_e s = 0$), positively selected (with selection coefficient $\gamma_+ > 0$), or negatively selected (with selection coefficient $\gamma_- < 0$) with probabilities p_0, p_+, and p_- (respectively). A likelihood ratio test (LRT) for positive selection is then performed by constraining the probability that a new mutation is advantageous to be zero under the null hypothesis (*i.e.*, $H_0 : \ p_+ = 0$ versus $H_1 : \ p_+ \neq 0$).

Table 1.1: Comparing the Average Number of Segregating Sites and Fixed Differences When Mutation Rates are Context-Dependent to their Expectations Under the Coalescent.

τ	$L\theta^a$		Segregating sites		Substitutions	
			$\rho^b = 0$	$\rho = 0.01$	$\rho = 0$	$\rho = 0.01$
10	2.15	total[c]	9.40 (4.11)	9.42 (3.42)	20.39 (5.12)	20.30 (4.71)
		obs.[d]	9.36 (4.05)	9.38 (3.41)	18.61 (4.70)	18.50 (4.35)
		exp.[e]	9.62 (4.12)	9.59 (3.66)	20.49 (5.24)	20.45 (4.83)
	4.3	total	18.78 (6.80)	18.82 (5.15)	40.89 (8.15)	40.83 (6.69)
		obs.	18.71 (6.71)	18.76 (5.13)	37.35 (7.46)	37.33 (6.26)
		exp.	19.31 (7.07)	19.26 (5.58)	40.94 (8.32)	41.00 (6.95)
	10.75	total	46.86 (14.83)	46.77 (8.44)	102.26 (16.25)	102.25 (10.89)
		obs.	46.68 (14.51)	46.60 (8.39)	93.63 (14.71)	93.61 (10.16)
		exp.	48.12 (15.25)	48.17 (9.57)	102.43 (16.68)	102.47 (11.23)
50	2.15	total	9.35 (3.98)	9.41 (3.46)	106.38 (10.44)	106.48 (10.46)
		obs.	9.33 (3.93)	9.37 (3.44)	82.45 (8.26)	82.40 (8.45)
		exp.	9.65 (4.14)	9.63 (3.67)	106.38 (10.62)	106.38 (10.49)
	4.3	total	18.68 (6.75)	18.81 (5.14)	212.88 (15.41)	212.78 (14.82)
		obs.	18.61 (6.64)	18.73 (5.11)	165.28 (12.31)	165.11 (11.98)
		exp.	19.23 (6.95)	19.24 (5.53)	213.00 (15.41)	212.94 (14.80)
	10.75	total	46.74 (14.57)	46.93 (8.44)	532.73 (26.45)	531.87 (23.83)
		obs.	46.58 (14.28)	46.77 (8.42)	414.42 (20.92)	413.95 (19.07)
		exp.	48.20 (15.48)	48.08 (9.55)	532.18 (26.68)	532.41 (23.42)

Note: For all simulations, $N_e = 250$ and $n = 50$. Parentheses indicate simulated SD.

[a] Average population scaled mutation rate per sequence ($2N_e\mu L$).
[b] Population scaled recombination rate between adjacent sites.
[c] Includes observed and unobserved mutations stored during simulation.
[d] Includes only observed mutations.
[e] Determined from coalescent simulations of the neutral model (Hudson, 2002).

To evaluate the robustness of this test for positive selection to ancestral misidentification, we simulated 1,000 datasets consisting of 200 independent loci with no intragenic recombination, each of length 1kb with θ per site = 0.0043 [similar to the value inferred by Williamson et al. (2005)], and an outgroup with population scaled divergence time of $\tau = 10$. For each pooled set of 200 loci, we sampled 50 sequences from one population and 2 sequences from the other, then constructed three SFS: the true SFS (using ancestral information stored during the simulation), the observed SFS (based on polarizing each SNP with homozygous sites in the outgroup sequence), and the corrected SFS based on applying our correction

[equation (1.7)] to the observed data. The LRT was then performed for each of the three SFS for all 1,000 datasets.

1.4 Results

The Effect of Ancestral Misidentification: Using forward simulations with a context-dependent mutation process, we compared the effect of ancestral misidentification on the SFS under neutrality to the effect of recurrent positive selection. Figure 1.2 shows that while the expected frequency distribution of neutral mutations (solid black line) differs from the frequency distribution of mutations with recurrent positive selection (solid grey line), ancestral misidentification can cause the frequency distribution of neutral mutations (dashed black line) to look substantially like positive selection. This qualitative similarity suggests that complex mutation patterns across sequences of finite length can compromise statistical tests of the standard neutral model, even when comparing species as closely related as human and chimpanzee.

Some of the most common methods for identifying departures from the standard neutral model (SNM) are based on comparing summary statistics of the SFS (Tajima, 1989; Fu and Li, 1993; Fay and Wu, 2000). Among the most common summaries are those that use the SFS to estimate the population scaled mutation rate, θ. One of the first estimates of θ was proposed by Watterson (1975), and uses the total number of segregating sites (θ_W). Another common estimate is based on the average pairwise heterozygosity [θ_π; Tajima (1983)]. Under the assumptions of the SNM, these two estimates of θ are unbiased, however, departures from the SNM will affect these two estimates differently. Tajima (1989) capitalized on this characteristic and proposed a statistical test for the equality of θ_W and θ_π, and defined the test statistic D. However, since neither θ_W nor θ_π require

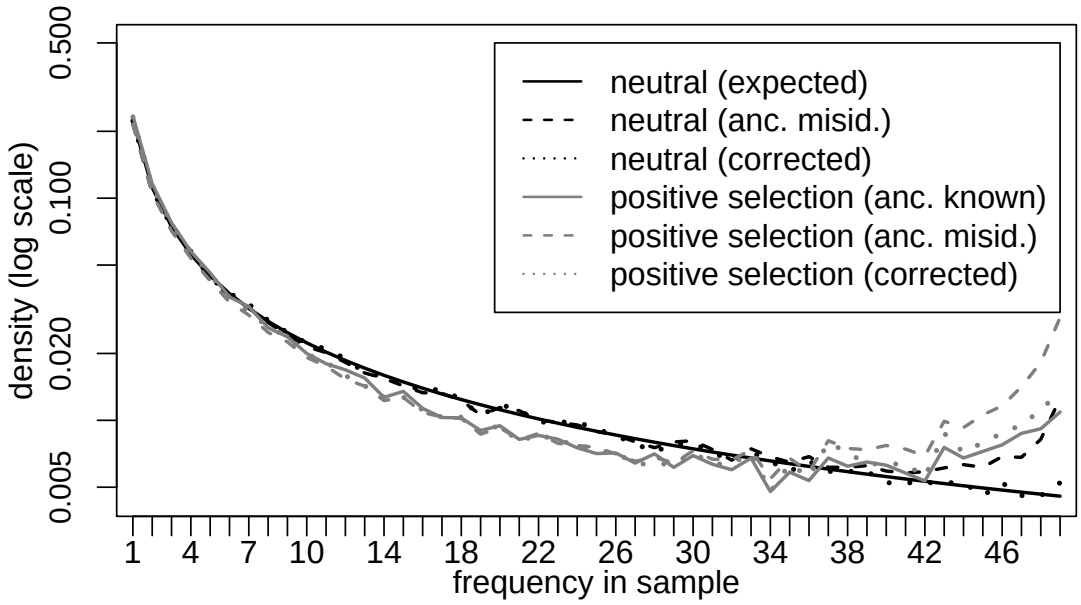

Figure 1.2: Simulation results showing the effect of ancestral misidentification on the SFS with and without rampant positive selection (grey and black, respectively) when using an outgroup as diverged as human-chimpanzee. Solid lines show the SFS when the ancestry for each SNP is known, dashed lines show the SFS when the ancestry for each SNP is inferred from the outgroup, and dotted lines show the result of correcting the observed data for ancestral misidentification. Each curve shows the average over 5,000 simulations, with an average population scaled mutation rate $\theta = 0.0043$ per site across sequences of length 1kb. In the case of positive selection all synonymous mutations are neutral ($\gamma = 2N_e s = 0$) and all nonsynonymous mutations are positively selected ($\gamma = 5$).

ancestral information (*i.e.*, both can be written as a function of the *folded* SFS), ancestral misidentification has no effect on Tajima's D. Indeed, the distribution of the D statistic for 5,000 forward simulations with ancestral misidentification significantly overlaps the distribution of D from 20,000 coalescent simulations for a range of parameters [MWU (Mann-Whitney U test) $P = 0.35$, 0.24, 0.39 for $\theta = 2.15, 4.3, 10.75$, respectively].

The second test statistic we considered was the statistic proposed by Fu and Li (1993) that uses outgroup information (denoted here as D_{fl}). The motivation for this test was based on the notion that purifying selection or a recent selective

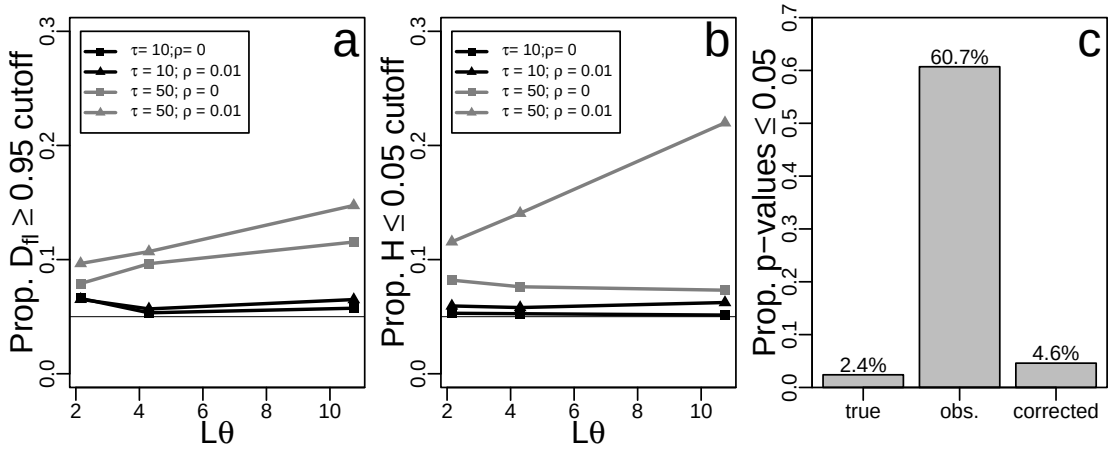

Figure 1.3: The effect of ancestral misidentification on statistical tests of the standard neutral model (SNM). The proportion of simulations resulting in (a) a Fu and Li (1993) D statistic (denoted D_{fl}) greater than the 0.95 critical value, (b) a Fay and Wu (2000) H statistic below the 0.05 critical value, and (c) a PRF-test for positive selection rejecting neutrality at the 5% level for the true, observed, and corrected data. For (a-b), the horizontal axis represents the average mutation rate across the simulated sequences with τ the population scaled divergence time and ρ the average per site recombination rate, and critical values determined using 20,000 coalescent simulations.

sweep would lead to a relative excess of young/rare mutations in the population, while balancing selection would lead to a relative deficiency. To capture this effect, D_{fl} statistically tests for the equality of θ_W and θ_1 (the estimate of θ based on the number of mutations that are carried by only a single chromosome in the sample). However, one of the effects of ancestral misidentification is that some rare mutations are mislabeled as being at high frequency. This causes Fu and Li's test to reject the SNM more frequently than expected at the 5% level due to a deficiency of rare mutants [*i.e.*, an excess of observations greater than the 95% critical value, shown in Figure 1.3(a)], especially when divergence times are long.

The third test statistic that we consider is Fay and Wu's H (Fay and Wu, 2000). This test was motivated by the desire to identify regions that may have been the target of a recent selective sweep. One characteristic of a selective sweep in a recombining locus is that some neutral alleles will have hitchhiked to very

high frequency in the population. To identify regions with an overrepresentation of high frequency mutants, H statistically tests for the equality of θ_π and θ_H [the estimate of θ based on the homozygosity of derived mutations; Fu (1995)]. However, one of the primary effects of ancestral misidentification is to increase the inferred proportion of derived mutations at very high frequency. As with the test of Fu and Li, ancestral misidentification can cause the H statistic to reject the SNM more often than expected at the 5% level [but due to an excess of large negative values that are below the 5% critical value, shown in Figure 1.3(b) and Baudry and Depaulis (2003)], especially when divergence times are long. Figure 1.3(b) also shows that ancestral misidentification has a larger impact on H in the presence of high recombination. This unintuitive result can be explained by Figure 1.4, which shows that while the mean value of H does not differ between $R = 0$ and $R = 25$ (Figures 1.4a versus 1.4b), the variance of H is much larger in the absence of recombination (indicated by comparing the 0.05 critical values). Figure 1.4(c) shows the effect of ancestral misidentification on H by plotting the value of H when all ancestral states are inferred from the outgroup minus the value of H when all ancestral states are known. These distributions overlap considerably (MWU $P = 0.167$), indicating that the effect of ancestral misidentification on H is independent of the recombination rate.

Correcting the SFS: To account for ancestral misidentification in the presence of context-dependent mutation rate variation, we model each component of the SFS as a mixture of SNPs whose ancestral states were correctly identified and SNPs whose ancestral states were misidentified (two unknown quantities; see Materials and Methods). In our model, the mixture components account for variation in substitution probabilities, stationary frequencies, and relative rates of mutation for each type of polymorphism in a given nucleotide context. The resulting model

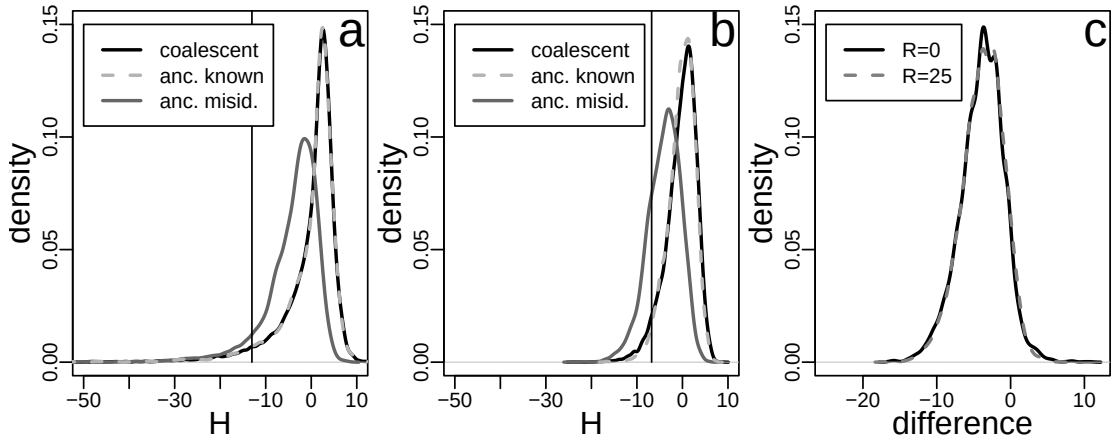

Figure 1.4: The distribution of the Fay and Wu (2000) H statistic for the average per sequence population scaled recombination rate $R = 0$ (a) and $R = 25$ (b) with the vertical lines indicating the 0.05 critical value obtained from 20,000 coalescent simulations (Hudson, 2002). (c) The distribution of the "difference" between H when the ancestral state of each polymorphism is inferred from the outgroup and H when the ancestral state is known (*i.e.*, saved during the simulation). Each curve represents 5,000 simulations of a sequence of length 2.5kb under the context-dependent mutation model discussed in the text with an average population scaled mutation rate per locus $\theta = 4N_e\mu = 10.75$.

is a linear system of equations with unknown quantities that can readily be solved to reconstitute the true SFS.

Figure 1.2 shows examples of simulated SFS with and without positive natural selection (in gray and black, respectively). The solid lines show the SFS when the ancestral state is known [or the expected SFS in the case of neutrality; Watterson (1975); Fu (1995)]. The dashed lines show the SFS when the ancestral state of each SNP is identified using an outgroup with divergence similar to human and chimpanzee. The dotted lines show the SFS after correcting for ancestral misidentification. If the correction were perfect, then the dotted lines would be directly on top of the solid lines. This is nearly the case for our neutral simulations, but a slight deviation remains in the case of positive selection. However, this result is promising, given that our correction does not explicitly model the effect of natural selection.

For a range of mutation, selection, and divergence parameters, 5,000 polymorphism datasets were simulated with an outgroup. In each dataset, the ancestral state of each SNP was identified using the outgroup, and the average proportion of SNPs whose ancestral states were misidentified was calculated (Figure 1.5). We then applied our correction to the observed data using both the observed divergence and the true divergence (measured in terms of the average number of substitutions per site), and calculated the number of SNPs whose ancestral states were still misidentified (Figure 1.5). Though our correction can eliminate a majority of the ancestral misidentification events for all the parameters tested (with and without selection), underestimating the level of divergence can have a strong impact (particularly for highly diverged outgroups). In our simulations of selection (Figures 1.2 and 1.5c-d), our correction tends to underestimate the effect of ancestral misidentification. This is primarily because selection provides an unaccounted for source of variance in the substitution probabilities across sites.

We determined the extent to which ancestral misidentification can produce statistical evidence for positive selection using a recently proposed statistical test based on the Poisson random field (Nielsen et al., 2005a). This test assumes that each nonlethal mutation enters the population and is assigned to one of three categories: neutral, positively selected, or negatively selected with probabilities p_0, p_+, and p_- (respectively). We found that the LRT based on the PRF framework is not robust to ancestral misidentification, but the false-positive rate can be controlled by correcting for ancestral misidentification. As shown in Figure 1.3(c), the LRT for the true SFS (based on saving the ancestral state of each polymorphism during the simulation) was conservative at the 5% level, rejecting H_0 only 2.4% of the time. Applying the LRT to the uncorrected (observed) SFS resulted in rejecting H_0 60.7% of the time at the 5% critical value. After performing the correction

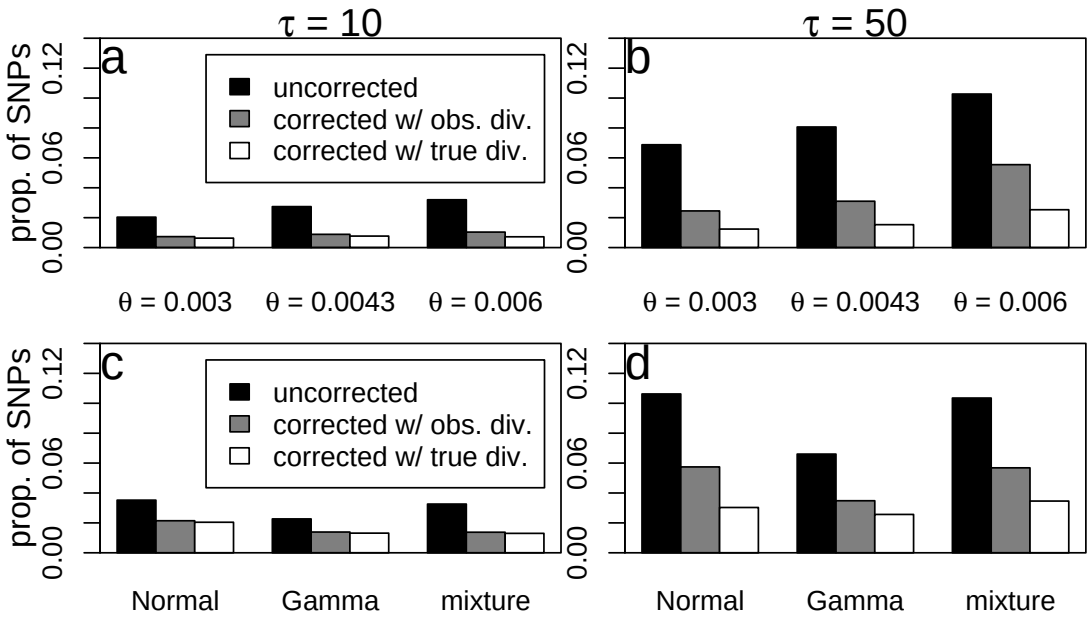

Figure 1.5: Simulation results showing the proportion of SNPs whose ancestral states have been misidentified for each value of the average population scaled mutation rate per site θ (a-b) or selection scheme (c-d). Each bar represents the average over 5,000 simulations. "Uncorrected" refers to inferring the ancestral state of each SNP using an outgroup sequence with population scaled divergence time $\tau = 10$ (a and c, representing human-chimpanzee divergence) or $\tau = 50$ (b and d, representing human-macaque divergence), and "corrected" refers to the data after performing our proposed correction with either the observed number of substitutions per site or the true number of substitutions per site (as determined by saving information during the simulation). Simulations with selection (c-d) were performed with $\theta = 0.0043$ per site and no intergenic recombination.

proposed in equation (1.7) using the observed divergence, the LRT rejected H_0 4.6% of the time at the 5% critical value.

An alternative approach to dealing with ancestral misidentification might be to restrict the analysis to the subset of SNPs that have outgroup support from multiple species (*e.g.*, the SNPs that have an allele matching both chimpanzee and rhesus macaque). To evaluate this strategy, we used a non-coding data set from the African American population [described by Hernandez et al. (2007c)]. Orthologous regions were identified in both chimpanzee and macaque using BLAT (Kent, 2002). As shown in Figure 1.7, using macaque as an outgroup to identify the ancestral

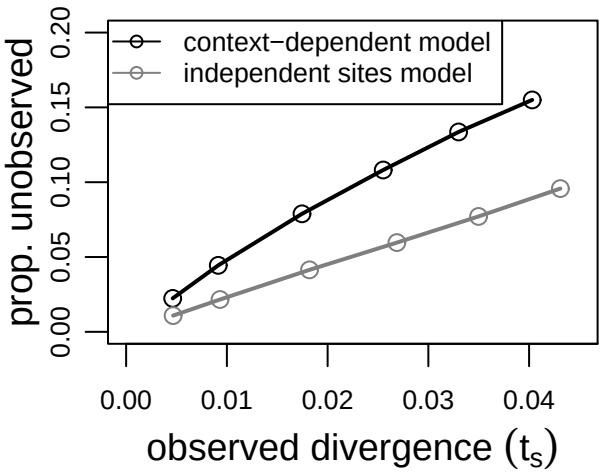

Figure 1.6: The proportion of substitutions that are unobserved as a function of the observed number of substitutions per site (τ_s) for the two mutation models discussed in the text.

states of human SNPs would lead to a very large excess of high frequency derived mutations (a result of ∼6% divergence over 25 million years). Using chimpanzee alone results in a lower proportion of high frequency derived mutations, but still an excess. Requiring the chimpanzee allele to match the macaque allele at each human SNP reduces the proportion of high-frequency derived mutations further, but there is still a slight excess. The excess high frequency tail completely vanishes when we apply the correction proposed above. It is also important to note that while the ancestral state of 10,179 SNPs could be identified using the chimpanzee, requiring the chimpanzee to equal the macaque results in just 7,082 SNPs (a reduction of 30%). However, by using the correction proposed in this paper when inferring ancestral states using chimpanzee, only 404 SNPs (3.96%) must be removed from the analysis (due to violations of the constant-context assumption).

To see the effect that the various strategies for identifying ancestral states of SNPs would have on population genetic analyses, we asked whether the SFS produced from each strategy could be explained by a simple demographic history with no selection. We used the method developed by Williamson et al. (2005),

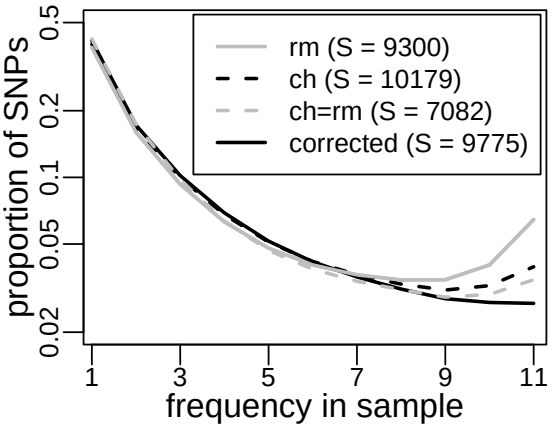

Figure 1.7: The SFS using four strategies to identify the ancestral states of each SNP for the observed data set. The solid gray (dashed black) line shows the result of using the rhesus macaque (chimpanzee) genome, the dashed gray line shows the subset of SNPs that have an allele matching both chimpanzee and rhesus macaque (ch=rm), and the solid black line shows the result of applying our correction to the observed data using only the chimpanzee outgroup. The legend indicates the resulting number of SNPs (S) for each strategy.

which fits a two-epoch demographic model to the SFS in the absence of selection. The model assumes that at some time τ in the past (scaled in terms of the current effective population size), the effective population size instantaneously changed from N_A to N_C (*i.e.*, from the ancestral to the current population size, with $\omega = N_A/N_C$ the magnitude of the change). We then performed a goodness-of-fit test, to see whether the inferred demographic model could sufficiently explain the data. Using the chimpanzee alone, we inferred $\tau = 0.11$ and $\omega = 0.49$ (that is, there was a two-fold population size increase). However, this simple demographic model cannot fully explain the data (as suggested by a goodness-of-fit $P = 1.1\times10^{-16}$, with twice the difference in log-likelihods $\Lambda = 92.9$ on eight degrees of freedom). Using both outgroups, we infer a slightly older and stronger demographic event, with $\tau = 0.14$ and $\omega = 0.42$. However, a goodness-of-fit test still rejects this model as a sufficient explanation for the data ($P = 2.6\times10^{-8}$, with $\Lambda = 51.0$). When we apply our correction to the data using chimpanzee as the only outgroup, we find the

demographic event was older than either of the other methods, and intermediate in strength, with $\tau = 0.19$ and $\omega = 0.47$. Importantly, a goodness-of-fit test on this data cannot reject the simple two-epoch demographic model ($P = 0.35$, $\Lambda = 8.9$).

1.5 Discussion

Ancestral misidentification of a SNP occurs when the ancestral state is not the allele observed at the orthologous site in the outgroup. Because most derived mutations tend to be rare (Watterson, 1975; Fu, 1995), ancestral misidentification most often leads to mislabeling low frequency derived mutations as very high frequency ones when constructing the site-frequency spectrum (SFS) using an outgroup. This results in an increase in the proportion of SNPs found at very high frequency, a pattern similar to positive selection. We have shown that ancestral misidentification causes many statistical tests to reject selective neutrality more frequently than expected.

Ignoring outgroup information and using the frequency distribution of minor alleles (the folded SFS) leads to a lack of power for some statistical tests (Bustamante et al., 2001). Moreover, the inclusion of another outgroup that is more diverged seems to cause a large fraction of the data (30%) to be removed while not significantly reducing the problem. We therefore propose a correction for the effect of ancestral misidentification on the SFS that takes advantage of our growing knowledge regarding the underlying neutral mutation process. The novelty of this method is in our use of a finite sites model for between species divergence, while maintaining an infinite sites model for within species polymorphism in the context of a realistic mutational process. Our technique is independent of population demography and ambivalent regarding the presence/absence of natural selection. Though it has long been suggested that frequency information is informative about

the ancestral state of a polymorphism (Watterson and Guess, 1977; Griffiths and Tavaré, 1998), we have modeled ancestral misidentification strictly as a function of the mutational process for the sake of being robust to unknown demographic histories, and to make inference regarding natural selection (or the distribution thereof) more conservative.

Our analysis shows that ancestral misidentification occurs frequently between human and chimpanzee, and we argue that this is primarily due to the fact that neutral mutation rates can vary both across sites as well as over time. An effect of ancestral misidentification could also be caused by sequencing errors in the outgroup. However, if this were the sole cause of the observed excess high frequency derived SNPs, then we would not expect to see a more dramatic effect when ancestral states were determined using rhesus instead of chimpanzee (Figure 1.7). Moreover, it may be the case that no species pair exists such that one can reliably polarize polymorphism data. For species more closely related than human and chimpanzee, further complications may arise due to lineage sorting and shared polymorphisms. It is therefore necessary to consider probabilistic models that can account for ancestral misidentification. However, our approach makes several simplifying assumptions that may be violated in some situations. For example, we assume that the flanking nucleotides of each polymorphism are constant between species, which may not always be the case. Using the polymorphism dataset consisting of 11,626 genes from 39 humans and a chimpanzee presented in Bustamante et al. (2005), we estimated that roughly 2.6% of SNPs in the coding regions of the human genome violate this constant context assumption (1.2% due to adjacent fixed differences and 1.4% due to adjacent SNPs). In the current non-coding dataset, 5.1% of SNPs were in violation of the constant context assumption (1.9% due to adjacent fixed differences and 3.2% due to adjacent polymorphisms). How-

ever, the inclusion and exclusion of polymorphic sites in violation of the constant context assumption has no effect on the qualitative results presented here.

The model presented here also assumes that all SNPs are independent, an assumption that is clearly not true for many SNPs occurring within the same gene. However, the effect of this assumption is minimized when pooling SNPs from across the genome. Moreover, because mutation rates and patterns are not necessarily constant across different regions of the genome (*e.g.*, it is unknown whether different isochores are following different mutational patterns), it is of interest to develop new techniques that will allow us to estimate context-dependent effects and account for ancestral misidentification on a local scale.

The effect we have observed will likely hold for any species pair in which there is sufficient mutation rate variation. In mammals, context-dependence contributes significantly to the amount of mutation rate variation. However, in some species, such as *Drosophila*, context-dependence has not been found (Andolfatto, 2005). In these species, other sources of mutation rate variation must be accounted for.

Though the context-dependent mutation pattern we have conditioned on may not be constant across all sequences, and though there may be other sources of mutation rate variation in addition to context-dependence, our analysis and simulations suggest a consistent pattern: without taking ancestral misidentification into account during the analysis of human polymorphism data, spurious signs of positive selection will be observed.

1.6 Acknowledgments

We are grateful to Jaroslaw Pillardy for computational assistance, Dara G. Torgerson for helpful comments, and Rasmus Nielsen for insightful suggestions. This work was funded by a National Science Foundation grant (0516310) to CDB.

CHAPTER 2

CONTEXT-DEPENDENT MUTATION RATES MAY CAUSE SPURIOUS SIGNATURES OF A FIXATION BIAS FAVORING HIGHER GC-CONTENT IN HUMANS*

*Originally published as: Hernandez, R. D., S. H. Williamson, L. Zhu, and C. D. Bustamante (2007). Context dependent mutation rates may cause spurious signatures of a fixation bias favoring higher gc-content in humans. *Mol Biol Evol*, 24(10):2196–2202, doi: 10.1093/molbev/msm149.

2.1 Abstract

Understanding the proximate and ultimate causes underlying the evolution of nucleotide composition in mammalian genomes is of fundamental interest to the study of molecular evolution. Comparative genomics studies have revealed that many more substitutions occur from G and C nucleotides to A and T nucleotides than the reverse, suggesting that mammalian genomes are not at equilibrium for base composition. Analysis of human polymorphism data suggests that mutations that increase GC-content tend to be at much higher frequencies than those that decrease or preserve GC-content when the ancestral allele is inferred via parsimony using the chimpanzee genome. These observations have been interpreted as evidence for a fixation bias in favor of G and C alleles due either to positive natural selection or biased gene conversion. Here, we test the robustness of this interpretation to violation of the parsimony assumption using a data set of 21,488 non-coding SNPs discovered by the NIEHS SNPs project via direct resequencing of $n = 95$ individuals. Applying standard non-parametric and parametric population genetic approaches we replicate the signatures of a fixation bias in favor of G and C alleles when the ancestral base is assumed to be the base found in the chimpanzee outgroup. However, upon taking into account the probability of misidentifying the ancestral state of each SNP using a context dependent mutation model, the corrected distribution of SNP frequencies for for GC-content increasing SNPs are nearly indistinguishable from the patterns observed for other types of mutations, suggesting that the signature of fixation bias is a spurious artifact of the parsimony assumption.

2.2 Introduction

Thirty years ago, mammalian genomes were first described as mosaics of isochores or long stretches of DNA with relatively homogeneous base composition (Macaya et al., 1976; Thiery et al., 1976). Regional variation in base composition is known to correlate with several complex biological processes. For example, regions with an excess of guanine and cytosine nucleotides (GC-rich regions) have been shown to have a lower density of LINE repeat elements (yet a higher density of Alu repeats), and higher levels of methylation, recombination, and gene density (Duret et al., 1995; Eyre-Walker, 1993; Fullerton et al., 2001; Jabbari and Bernardi, 1998; Lander et al., 2001; Mouchiroud et al., 1991; Smit, 1999). Isochores appear to have entered vertebrate genomes ∼310–350 million years ago (Bernardi et al., 1997), but there is still considerable debate regarding their formation and which evolutionary forces are acting to maintain them (Bernardi, 2000; Fryxell and Zuckerkandl, 2000; Meunier and Duret, 2004).

Comparative and population genomic data suggests that mammalian genomes may not be at compositional equilibrium, and predict that GC-rich isochores are being degraded by mutation (Galtier and Gouy, 1998; Arndt et al., 2003; Belle et al., 2004; Meunier and Duret, 2004). Likewise, several studies have analyzed human polymorphism data, and found evidence for a fixation bias in favor of mutations that increase GC-content [*i.e.*, from A or T to G or C, denoted AT→GC; Eyre-Walker (1999); Webster et al. (2003)]. Such a fixation bias could be caused by either natural selection or biased gene conversion (*i.e.*, when nucleotide mismatches formed from the hybridization of two DNA strands during meiosis is repaired asymmetrically). Thus the prevailing view is that mutation biases or compositional disequilibrium tend to erode GC-content and biased fixation rates tend to increase it.

A powerful approach for detecting a fixation bias is the analysis of the frequency distribution of SNPs (*i.e.*, the "unfolded" site frequency spectrum, or SFS) (Akashi, 1999; Bustamante et al., 2001; Nielsen et al., 2005b). Several studies have utilized such tools with a variety of models and data summaries to suggest the presence of an AT→GC fixation bias in the human genome, especially in regions of high GC-content (Duret et al., 2002; Lercher et al., 2002; Webster and Smith, 2004), with similar patterns observed in the proximal regions of recombination hotspots (Spencer, 2006; Spencer et al., 2006). Many of these tests rely on using a parsimony assumption and outgroup sequence data to distinguish ancestral from derived alleles (*i.e.*, the segregating allele matching the outgroup allele is assumed to be ancestral), though Lercher et al. (2002) did not use an outgroup and Webster and Smith (2004) developed a weighted parsimony technique (discussed below).

In a companion article (Hernandez et al., 2007b), we present a flexible method for relaxing the parsimony assumption by using a context-dependent mutation model which includes features such as elevated mutation rates at CpG dinucleotides, increased propensity for transitional versus transversional mutations, as well as other directional and contextual mutation biases inferred along the human lineage by Hwang and Green (2004). We found that even for species as closely related as human and chimpanzee, enough unobserved nucleotide substitutions could have occurred to make some population genetic analyses spuriously reject neutrality. The spurious signal of selection is due to misidentifying the ancestral state of some SNPs via the parsimony assumption. Since most derived mutations tend to be rare, ancestral misidentification will most often lead to mislabeling low frequency variants as extremely high frequency mutations (a characteristic that would be consistent with a fixation bias).

Because the mutation rate from GC→AT tends to be approximately two-fold higher than the reverse (Hwang and Green, 2004), if an AT→GC substitution should occur during the divergence of two species, the site-specific mutation rate would immediately increase ∼two-fold, thereby doubling the relative probability of another mutation at this site on the same lineage. Should a polymorphism arise at such a site, an allele that matches the outgroup would be due to homoplasy, and not indicative of ancestry. A further complicating factor in the analysis of the SFS is that historical demographic effects can have a large impact on the underlying frequency distribution of derived mutations (Slatkin and Hudson, 1991; Nielsen, 2001). Without explicitly accounting for such demographic forces, population genetic tests of the SFS can either lead to a false rejection of selective neutrality or to a poor fitting model.

Here, we use two approaches to test the fixation bias hypothesis using a large non-coding human polymorphism data set with and without correcting for ancestral misidentification. We find that when ancestral misidentification is not taken into account, neutral population genetic models tend to fit the data very poorly and suggest strong evidence for non-neutral processes acting on GC-content in the human genome. After correcting for ancestral misidentification using the method of Hernandez et al. (2007b), we find much of the statistical evidence for a fixation bias favoring G and C alleles from SNP data goes away, suggesting that the result is an artifact of ancestral misidentification.

2.3 Materials and Methods

2.3.1 Data

The data used in this study were retrieved from the NIEHS Environmental Genome Project website (`http://egp.gs.washington.edu`). Our final dataset represents a collection of SNPs obtained through direct sequencing of 161 genes (along with flanking and intronic regions) in a sample of 95 individuals (190 chromosomes) from 5 worldwide populations (panel 2): 15 African American, 12 African (Yoruba), 22 European, 22 Hispanic, and 24 Asian individuals (Livingston et al., 2004). Sixteen genes had less than 1kb of non-coding sequence, and were removed from the analysis (a list of genes used can be obtained from the corresponding author). Orthologous chimpanzee sequences were obtained using BLAT (Kent, 2002) on build 1 of the chimpanzee genome (Chimpanzee Sequencing and Analysis Consortium, 2005). To maximize the outgroup coverage of our human sequence data, we divided long sequences into segments of length 25 kb with 2 kb of overlap. We then used BLAT on each segment against the chimpanzee genome. This resulted in 83% of human nucleotide bases having outgroup information. The non-coding portion of the dataset spans 7.5 Mb, and includes 21,488 SNPs. Some SNPs were removed from the analysis because they had missing outgroup/context information (1,901), they were adjacent to another SNP (678), they were in violation of the constant context assumption (646), or the chimpanzee allele did not match one of the segregating alleles (163). Our final non-coding dataset includes 16,866 SNPS from flanking, untranslated (UTR), and intronic regions.

Our analysis is based primarily on the frequency distribution of derived mutations at all observed SNPs (*i.e.*, the site-frequency spectrum, or SFS). The SFS is a random vector that represents the number of SNPs whose derived allele is

observed at each frequency in our sample of chromosomes. Missing sequence data from some chromosomes can cause SNPs to have a sample size smaller than the total set of 190 chromosomes. Rather than discard all such SNPs, we only removed SNPs that had a sample size less than 40 (four SNPs), and performed our analysis on the expected SFS in a subsample of size 40 chromosomes (Marth et al., 2004; Nielsen et al., 2004).

Below we describe a population genetic model to infer the parameters of a demographic model that allows for a fixation bias favoring GC-content. Because our demographic model cannot accommodate the complex dynamics of the full dataset, only the results of analyzing the African-American and Yoruban populations will be reported (though not shown, the model fits the other populations very poorly). To accommodate the missing data in the African American and Yoruban populations, both were analyzed using the expected SFS in a subsample of size 12 chromosomes (as above).

2.3.2 Testing the Significance of a Fixation Bias Favoring GC-Content

Our interest is in identifying whether or not natural selection or biased gene conversion has been acting on GC-content in the human genome. To do so, we analyzed the dataset pooled across populations using two non-parametric tests: the Mann-Whitney U test, or MWU, and the Kolmogorov-Smirnov test, or KS. We also analyzed the African-American and Yoruban populations individually using a population genetic model of demography and selection. We performed all tests before and after correcting for the probability of misidentifying the ancestral state of each SNP [as discussed by Hernandez et al. (2007b)].

Because recent demographic effects can confound inference of fixation biases using the SFS, applying population genetic techniques to infer the presence/strength of a fixation bias without taking into account the effect of demography may lead to highly biased results (Williamson et al., 2005; Nielsen, 2001). We therefore adapted a recently proposed method for simultaneously inferring the demographic history of a population and the strength of a fixation bias for the analysis of both the African American and Yoruban populations (Williamson et al., 2005). In its original form, the method pooled synonymous and non-coding SNPs (*i.e.*, the putatively neutral, or class 1 SNPs) to estimate the time back to a population size change event (t_{dem}) which had magnitude $\omega = N_a/N_c$ (the ratio of the ancestral to current population sizes). Then, the strength of selection acting on nonsynonymous SNPs (*i.e.*, class 2) was inferred conditional on the non-stationary demographic model from synonymous and non-coding SNPs.

We consider four models of the SFS. The first model (M_{SNM}) represents the standard neutral model (SNM), and has zero free parameters. The second model (M_{dem}) represents a neutral demographic model, and has two free parameters (t_{dem} and ω). The third model (M_{fix}) assumes that the size of the population changed at some time t_{dem} in the past with magnitude ω, and that all mutations are selectively neutral except for some proportion of AT→GC mutations (π_γ), which experience a common fixation bias denoted γ (a total of four free parameters). In this model, we consider the potential fixation bias favoring AT→GC mutations to be analogous to the population scaled selective effect of a new mutation as in previous studies (Duret et al., 2002; Lercher et al., 2002; Webster and Smith, 2004), which could either be due to natural selection or biased gene conversion.

Model M_{fix} is a modification of Williamson et al. (2005), which allows only a proportion (π_γ) of non-lethal mutations to be subject to a fixation bias. Allowing

only a proportion of AT→GC mutations to be subject to the fixation bias enables us to identify the effect even if it is restricted to small regions of the genome [*e.g.*, regions of high GC-content (Duret et al., 2002; Lercher et al., 2002) or near recombination hotspots (Spencer, 2006; Spencer et al., 2006)]. To write down the likelihood function for the new model, we updated the distribution of allele frequencies for the AT→GC mutations [$f_2(\cdot)$ in the notation of Williamson et al. (2005)]. In our model, we assume that the fixation bias parameter of a non-lethal AT→GC mutation is either neutral (*i.e.*, $\gamma = 0$) or non-neutral (*i.e.*, $\gamma \neq 0$), with probabilities 1-π_γ and π_γ (respectively), and that all other types of mutations drift neutrally (*i.e.*, in the absence of a fixation bias, $\gamma = 0$). This implies that the fixation bias of non-lethal AT→GC mutations come from a mixture distribution, which can readily be incorporated into the new distribution of allele frequencies, $\phi(x \mid \gamma, \pi_\gamma, t_{dem}, \omega)$. Namely,

$$\phi(x \mid \gamma, \pi_\gamma, t_{dem}, \omega) = \pi_\gamma f_2(x \mid \gamma, t_{dem}, \omega) + (1 - \pi_\gamma) f_2(x \mid 0, t_{dem}, \omega), \qquad (2.1)$$

where $f_2(x \mid \gamma, t_{dem}, \omega)$ derives from the numerical solution to the allele frequency distribution of a mutation with a fixation bias of strength γ in a non-stationary population found by Williamson et al. (2005). The probability of observing an AT→GC mutation at frequency i is $P(i \mid \gamma, \pi_\gamma, t_{dem}, \omega)$, which can be found by substituting our equation (2.1) into equation [9] of Williamson et al. (2005). Since we assume that all mutations that are not from AT→GC drift neutrally, the probability of observing a non-AT→GC mutation at frequency i is $P(i \mid 0, 0, t_{dem}, \omega)$, which is equivalent to equation [6] of Williamson et al. (2005). The likelihood function for this model, $\mathcal{L}_{fix}(\gamma, \pi_\gamma, t_{dem}, \omega)$, is then written as

$$\mathcal{L}_{fix}(\gamma, \pi_\gamma, t_{dem}, \omega) = \prod_{i=1}^{n-1} P(i \mid \gamma, \pi_\gamma, t_{dem}, \omega)^{K_{\mathrm{AT}\to\mathrm{GC}}(i)} \prod_{i=1}^{n-1} P(i \mid 0, 0, t_{dem}, \omega)^{K_{other}(i)} \qquad (2.2)$$

where $K_{\mathrm{AT}\to\mathrm{GC}}(i)$ is the number of SNPs from AT→GC at frequency i, and $K_{other}(i)$ is the number of SNPs at frequency i that either preserve or decrease

GC-content. Note that we did not implement the probability of ancestral misidentification used in Williamson et al. (2005) for any of our likelihood calculations, and that the log-likelihood of this model is denoted $L_{\text{fix}} = \log(\mathcal{L}_{\text{fix}})$. For inference, we optimize L_{fix} across all four parameters simultaneously, as compared to Williamson et al. (2005), which inferred the selective effect conditional on the inferred demographic history.

The fourth model (M_{mult}) is a multinomial model, where the probability of observing a SNP of a given mutation class (*i.e.*, AT→GC or other) at frequency i is given by the observed proportion of SNPs in that mutation class at frequency i. This is the most general model, and has $2(n-2)$ free parameters.

Our test for a fixation bias favoring GC-content involves four likelihood ratio tests (LRTs). The first test compares model M_{SNM} to model M_{dem}. If the log-likelihood of M_{dem} (denoted L_{dem}) is significantly larger than L_{SNM}, we reject M_{SNM} in favor of the neutral demographic model. Our second test compares the log-likelihood of the neutral demographic model (L_{dem}) to the log-likelihood of the demographic model with a fixation bias favoring AT→GC mutations (L_{fix}). If L_{fix} is significantly larger than L_{dem}, we reject the neutral demographic hypothesis in favor of the model with a fixation bias favoring AT→GC mutations. Finally, we perform a goodness of fit test on both M_{dem} and M_{fix} by comparing L_{dem} and L_{fix} to L_{mult} (the log-likelihood of model M_{mult}). Note that a p-value larger than 0.05 for a goodness of fit test indicates that the model under consideration sufficiently explains the data.

For all likelihood ratio test, p-values were estimated from 2,000 coalescent simulations of the LRT statistic. In order for our simulations to mimic the true data as much as possible, we accounted for the inferred demographic history of each population, linkage among sites, mutation rate variation, and the distribution of

missing data that we observed. We first estimated the demographic parameters for each population independently (using model M_{dem}). To estimate the population scaled recombination rate (R), we applied a novel approach proposed by Zhu, Feng and Bustamante (in review). This method uses the variances and co-variances of unphased SNPs at different frequencies to predict the local recombination rate by multiple linear regression and non-parametric bootstrap resampling. For the data in this paper, we first fit the regression model by simulating 1,000 replicate datasets under the inferred demographic model for each gene region in each population using the coalescent with R in the range $\{1, 2, 5, 10, 20, 50, 100, 200, 400, 1000\}$. Each replicate has the same number of sequences (n) and segregating sites (S) as in the observed data. We estimate the variances of the site-frequencies for each replicate by non-parametric bootstrapping and use the mean of the variances over 1000 replicates to fit the best linear regression on R. R-squares of the linear models are all above 90%. We then bootstrap to estimate the variances of the site-frequencies for the observed gene region and use the linear relationship between the log of the variances and log R to predict the local recombination rate for each gene region. For gene regions with less than ten SNPs, the recombination rate was assumed to be zero.Our estimate of the mutation rate for each gene region (independent for each population) was based on the observed number of segregating sites and the inferred demographic history.

After generating 2,000 coalescent simulations for each gene region in each population, we randomly assigned some SNPs to be of type AT→GC based on the proportion of SNPs that were observed to be of that type in our data. To account for the observed pattern of missing data, each simulated SNP was assigned to a new sample size according to the proportion of SNPs observed at each sample size. The frequency of the derived state in the reduced sample size follows a hyperge-

ometric distribution. That is, if the frequency of the derived state of a SNP was i in the original sample size of n chromosomes, then the probability that j copies ($j \leq i$) were observed in the reduced sample of size n' chromosomes was $\frac{\binom{i}{j}\binom{n-i}{n'-j}}{\binom{n}{n'}}$.

After generating the missing data for each of the coalescent simulations, we generated the expected SFS in a subsample of size 12 chromosomes using the same technique as in the observed data. Finally, we performed the LRTs described above on each coalescent simulation to approximate the distribution of the LRT statistic, from which we obtained our p-values.

2.4 Results and Discussion

Under the assumption of parsimony, we identified the ancestral state of each SNP in our non-coding dataset using the chimpanzee genome (see section Data). We refer to this dataset as the uncorrected dataset. Shown in Figure 2.1(a) are the normalized SFS for SNPs that decrease GC-content (*i.e.*, GC→AT), increase GC-content (*i.e.*, AT→GC), and preserve GC-content (*i.e.*, other) for the uncorrected dataset (pooled into bins of size 3). We found that while there is no statistical evidence that the frequency distribution of GC→AT mutations differs from GC-content preserving mutations (p-value=0.788 MWU; p-value>0.99 KS), the SFS for AT→GC mutations is significantly different from both GC→AT mutations (p-value=2.04×10^{-07} MWU; p-value=0.0004 KS) and GC-content preserving mutations (p-value = 6.36×10^{-05} MWU; p-value = 0.0100 KS).

However, since the uncorrected SFS were generated using orthologous chimpanzee sequences, ancestral misidentification of some SNPs could have occurred. We applied a correction for ancestral misidentification (Hernandez et al., 2007b) using the observed non-coding divergence of 0.012 substitutions per site. Figure 2.1(b) shows the corrected SFS for the three classes of mutations discussed

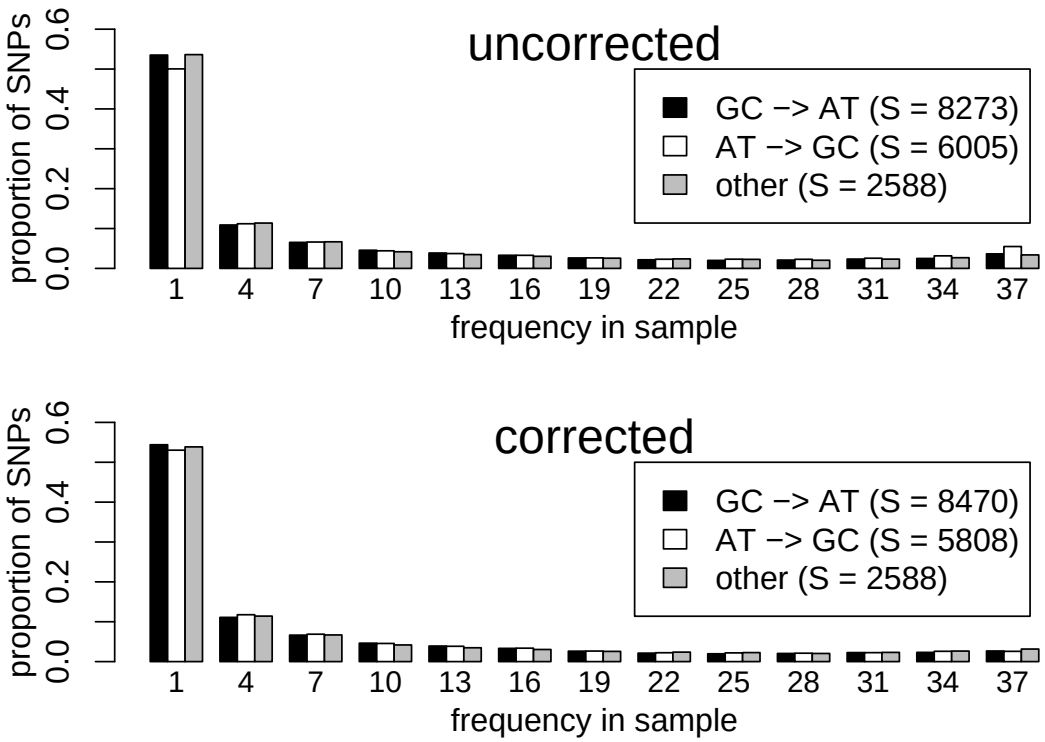

Figure 2.1: SFS for SNPs that decrease GC-content (*i.e.*, GC→AT), increase GC-content (*i.e.*, AT→GC), and preserve GC-content (*i.e.*, other, referring to A↔T or G↔C) in the non-coding data discussed in the text before (a) and after (b) correcting for ancestral misidentification using the observed non-coding divergence of 0.012 substitutions per site between human and chimpanzee (pooled into frequency bins of size three for visualization purposes only). In parentheses are the number of SNPs in each category (S).

above, and Table 2.1 shows the average frequency of a derived mutation before and after correcting for ancestral misidentification. We found that there was a net change in the classification of 197 SNPs initially observed to increase GC-content (≈3.3%). These SNPs may actually have been GC-content decreasing SNPs, but due to ancestral misidentification, the orientation was swapped. After correcting for ancestral misidentification, we find no statistical evidence suggesting that the SFS for AT→GC SNPs differ from either GC→AT SNPs (p-value=0.0967 MWU; p-value=0.3260 KS) or GC-content preserving SNPs (p-value=0.5541 MWU; p-

Table 2.1: Average frequencies of putatively derived mutations.

	AT→GC		GC→AT	
pop.	unc.	cor.	unc.	cor.
All	0.244	0.207	0.211	0.200
Af. Am.	0.291	0.263	0.272	0.262
Yor.	0.301	0.272	0.265	0.256

value>0.99 KS). This suggests that ancestral misidentification is an alternative explanation for the observed deviation of AT→GC polymorphisms from other types of polymorphisms.

Previous studies have fit population genetic models to observed SFS to assess the statistical evidence for positive selection (or biased gene conversion) acting on GC-content in the human genome using a chimpanzee outgroup [Duret et al. (2002) and Webster and Smith (2004); Lercher et al. (2002) also fit a population genetic model to a variant of the SFS that does not use an outgroup]. However, these studies have not accounted for the effect of historical population size changes (which may confound inference regarding fixation biases), or sufficiently addressed the effect of ancestral misidentification. We apply a population genetic model that accounts for both of these complications, but because our simple demographic model does not fit the Asian and Caucasian populations well (not shown), we focus on the analysis of the African American and the Yoruban datasets.

We first tested whether this non-coding dataset showed evidence for a non-stationary demographic history using an adapted version of the numerical technique developed by Williamson et al. (2005). This method assumes that the population experienced an instantaneous size change from the ancestral size of N_a to the current size N_c (where $\omega = N_a/N_c$ denotes the magnitude of the change) at a time t_{dem} in the past. Table 2.2 shows the parameter estimates of the neutral demographic model (M_{dem}), and Table 2.3 shows that the SNM can clearly be rejected both before and after correcting for ancestral misidentification in both

Table 2.2: Parameters and likelihoods for population genetic models.

	Pop.	model	$\widehat{t_{dem}}^a$	$\widehat{\omega}^b$	$\widehat{\gamma}^c$	$\widehat{\pi_\gamma}^d$	-L^e
uncorrected	Af. Am.	M_{SNM}	-	-	-	-	19106.9
		M_{dem}	0.11	0.48	-	-	18983.3
		M_{fix}	0.13	0.44	25.7	0.12	18951.4
		M_{mult}	-	-	-	-	18929.9
	Yor.	M_{SNM}	-	-	-	-	17652.9
		M_{dem}	0.16	0.55	-	-	17557.9
		M_{fix}	0.25	0.51	490.0	0.07	17514.6
		M_{mult}	-	-	-	-	17492.6
corrected	Af. Am.	M_{SNM}	-	-	-	-	18705.8
		M_{dem}	0.19	0.47	-	-	18518.1
		M_{fix}	0.19	0.47	18.4	0.01	18517.9
		M_{mult}	-	-	-	-	18510.2
	Yor.	M_{SNM}	-	-	-	-	17271.7
		M_{dem}	0.30	0.51	-	-	17113.0
		M_{fix}	0.32	0.48	0.50	1.0	17108.7
		M_{mult}	-	-	-	-	17096.1

[a]Population scaled time back to demographic event.
[b]Magnitude of population size change (ancestral/current).
[c]Population scaled selection coefficient.
[d]Proportion of non-lethal AT→GC mutations that have a selective effect.
[e]Minus log-likelihood of the model.

populations.

We then tested for a fixation bias favoring AT→GC mutations by extending the model of Williamson et al. (2005) to a model that allows only a proportion ($0 \leq \pi_\gamma \leq 1$) of non-lethal AT→GC mutations to have a fixation bias with strength γ, while the remaining proportion of AT→GC mutations (as well as the other mutation classes) drift neutrally (see Materials and Methods). This model implicitly accounts for the possibility of a fixation bias that only acts in confined regions of the genome (*e.g.*, in GC-rich regions or near recombination hotspots). Table 2.2 shows the parameter values obtained from each of the population genetic models

Table 2.3: P-values for the LRTs of fixation bias for GC-content before and after correcting for ancestral misidentification.

pop.	test	uncorrected	(corrected)[a]
Af.Am.	M_{SNM} vs. M_{dem}	**<0.0005**	(**<0.0005**)
	M_{dem} vs. M_{fix}	**<0.0005**	(0.64)
	GOF(M_{dem})	**<0.0005**	(0.23)
	GOF(M_{fix})	**0.0045**	(0.19)
Yor.	M_{SNM} vs. M_{dem}	**<0.0005**	(**<0.0005**)
	M_{dem} vs. M_{fix}	**<0.0005**	(**0.019**)
	GOF(M_{dem})	**<0.0005**	(**0.038**)
	GOF(M_{fix})	**0.017**	(0.096)

[a]Result of the test after correcting for ancestral misidentification in parentheses.

we evaluated, and Table 2.3 shows the p-values obtained via simulation for each LRT. Before correcting for ancestral misidentification, we can clearly reject the neutral demographic model in both populations. However, in both populations, a goodness of fit test narrowly rejects model M_{fix}, suggesting that it cannot fully explain the data.

After correcting for ancestral misidentification, the evidence for a fixation bias favoring AT→GC mutations in the human genome nearly vanishes. In the African American population, we cannot statistically reject the neutral demographic model in favor of a model allowing for a fixation bias (p=0.643, Table 2.3). Moreover, a goodness of fit test of the neutral demographic model in this population suggests that the neutral demographic model is sufficient to explain the data (p=0.234, Table 2.3).

The data for the Yoruban population is slightly more complicated. After correcting for ancestral misidentification, the neutral demographic model is narrowly rejected at the 0.05 significance level (p=0.0192, Table 2.3), suggesting that there is either slight evidence for an extremely weak fixation bias ($\gamma = 0.5$) acting on

all AT→GC mutations, or that the simple 2-epoch demographic model is insufficient. Goodness of fit tests suggest that while the neutral demographic model can narrowly be rejected at the 0.05 significance level (p=0.0383), there is marginal support for the model that allows for a fixation bias (p=0.0963).

2.5 Conclusion

We found that after correcting for ancestral misidentification, much of the evidence for the fixation bias favoring G and C alleles disappeared. This is because ∼3.3% of SNPs identified to be of type AT→GC (most of which were at very high frequency) may actually have been of type GC→AT (and at very low frequency). Such an effect might be expected since the overall mutation rate from GC→AT tends to be roughly twice as large as the rate from AT→GC along primate lineages (Hwang and Green, 2004). That is, if the allele ancestral to human and chimpanzee were an A or T, and an AT→GC substitution occurred along the human lineage, then the mutation rate at this site would, on average, double. This would, in turn double the probability of subsequently sampling a polymorphism at the same site but with a misidentified frequency based on the simple parsimony assumption. We therefore conclude that much of the evidence for a recent fixation bias favoring GC-content in humans based on population genetic data may be a result of failing to account for multiple hits at rapidly evolving sites between humans and chimpanzees. However, a previous study developed a weighted parsimony method to account for ancestral misidentification of human SNPs using a chimpanzee outgroup without accounting for context-effects (Webster and Smith, 2004). Interestingly, an excess of AT→GC SNPs at very high frequency remained after their correction. Given that the SFS we have observed in this data set is consistent with our neutral simulations of ancestral misidentification (Hernandez

et al., 2007b), it seems as though the weighted parsimony method was unable to fully correct for ancestral misidentification.

We emphasize that eliminating hypermutable CpG sites from consideration in SNP studies is not sufficient to safeguard against this effect, nor is restricting the analysis to those SNPs that have outgroup support from multiple species. Both of these techniques tend to require much of the data to be discarded (thereby leading to an ascertainment bias) without guaranteeing against further ancestral misidentification (Hernandez et al., 2007b). Rather, we recommend employing a parametric model for the data that can account for uncertainty in the ancestral states of all SNPs as well as mutation rate heterogeneity, since this approach both theoretically and in simulations appears to have proper type I (false positive) error rates (Hernandez et al., 2007b).

2.6 Acknowledgments

We are grateful to D. G. Torgerson for helpful comments and assistance with obtaining orthologous chimpanzee sequences, and two reviewers that provided very helpful comments. This work was funded by a National Science Foundation grant (0516310) to CDB.

CHAPTER 3

DEMOGRAPHIC HISTORIES AND PATTERNS OF LINKAGE DISEQUILIBRIUM IN CHINESE AND INDIAN RHESUS MACAQUES*

*Originally published as: Hernandez, R. D., M. J. Hubisz, D. A. Wheeler, D. G. Smith, B. Ferguson, J. Rogers, L. Nazareth, A. Indap, T. Bourquin, J. McPherson, D. Muzny, R. Gibbs, R. Nielsen, and C. D. Bustamante (2007). Demographic histories and patterns of linkage disequilibrium in Chinese and Indian rhesus macaques. *Science*, 316(5822):240–243, doi: 10.1126/science.1140462.

3.1 Abstract

To understand the demographic history of rhesus macaques (Macaca mulatta) and document the extent of linkage disequilibrium (LD) in the genome, we partially resequenced five ENCyclopedia of DNA Elements regions in 9 Chinese and 38 captive-born Indian rhesus macaques. Population genetic analyses of the 1467 single-nucleotide polymorphisms discovered suggest that the two populations separated about 162,000 years ago, with the Chinese population tripling in size since then and the Indian population eventually shrinking by a factor of four. Using coalescent simulations, we confirm that these inferred demographic events explain a much faster decay of LD in Chinese ($r^2 \approx 0.15$ at 10 kilobases) versus Indian ($r^2 \approx 0.52$ at 10 kilobases) macaque populations.

3.2 Introduction

Rhesus macaques (*Macaca mulatta*) and humans shared a most recent common ancestor (MRCA) ∼25 million years ago (Ma), and our genomes differ at <7% of nucleotide bases (Rhesus Macaque Genome Sequencing and Analysis Consortium, 2007). Rhesus and humans, therefore, share a large number of fundamental biological characteristics, including many underlying genetic and physiological processes that lead to disease. For this reason, rhesus macaques have become a model organism for vaccine research (Weiss, 2001; Ling et al., 2002) as well as studies of normal human physiology and disease. Although previous studies of genetic variation in rhesus have described >300 microsatellite polymorphisms (Rogers et al., 2006; Raveendran et al., 2006), identifying specific genetic risk factors for disease requires a much greater resolution of genetic variation across the genome.

The current geographic range of rhesus macaques is larger than any other non-human primate, stretching from western India and Pakistan to the eastern shores of China (Figure 3.1). Fossil records suggest that the genus *Macaca* originated in northern Africa approximately 5.5 Ma, followed by migration through the Middle East and into northern India by ∼3 Ma (Delson, 1980). By ∼2 Ma, macaques had traversed most of China and reached the Indonesian archipelago, where the putative ancestral species of rhesus macaque, *M. fascicularis*, is thought to have originated (Delson, 1980; Abegg and Thierry, 2002).

Previous studies of mitochondrial DNA (Smith and McDonough, 2005), major histocompatibility complex (MHC) alleles (Viray et al., 2001), and single-nucleotide polymorphisms (SNPs) in gene-linked regions (Ferguson et al., 2007) suggest moderate levels of genetic differentiation between captive-born Indian and Chinese rhesus populations. Developing a more thorough understanding of genetic variation within and between these two populations has important implications for biomedical research. For example, when infected with the simian immunodeficiency virus, animals from Chinese populations develop AIDS-like symptoms more slowly than animals from Indian populations (Ling et al., 2002).

3.3 Results and Conclusions

We have identified 1476 SNPs by sequencing >150 kb of DNA across 5 ENCyclopedia of DNA Elements [ENCODE; see Appendix A, ENCODE Project Consortium (2004)] regions located on separate autosomal chromosomes in nine captive-born from wild-caught Chinese and 38 captive-born Indian rhesus macaques. The Chinese animals derive from three distinct geographical sites, whereas the Indian animals came from three different colonies in the United States (Figure 3.1). Individuals were chosen to represent rhesus macaque populations that are currently being

studied by the international community and to minimize relatedness (with most individuals in the study being unrelated back to the founding of the colony into which they were born, and none having a shared grandparent; see Appendix A). In our sample of 1476 SNPs discovered, only 486 (33%) were shared across both populations, whereas 604 were found only in the Chinese population (61% of 1090 SNPs observed) and 386 were found only in the Indian population (39% of 872 SNPs observed). The frequency distribution of derived mutations across SNPs (using DNA sequence from the ENCODE project for baboon, *Papio cynocephalus anubis*, to infer the putative ancestral states; see Appendix A) shows that the Chi-

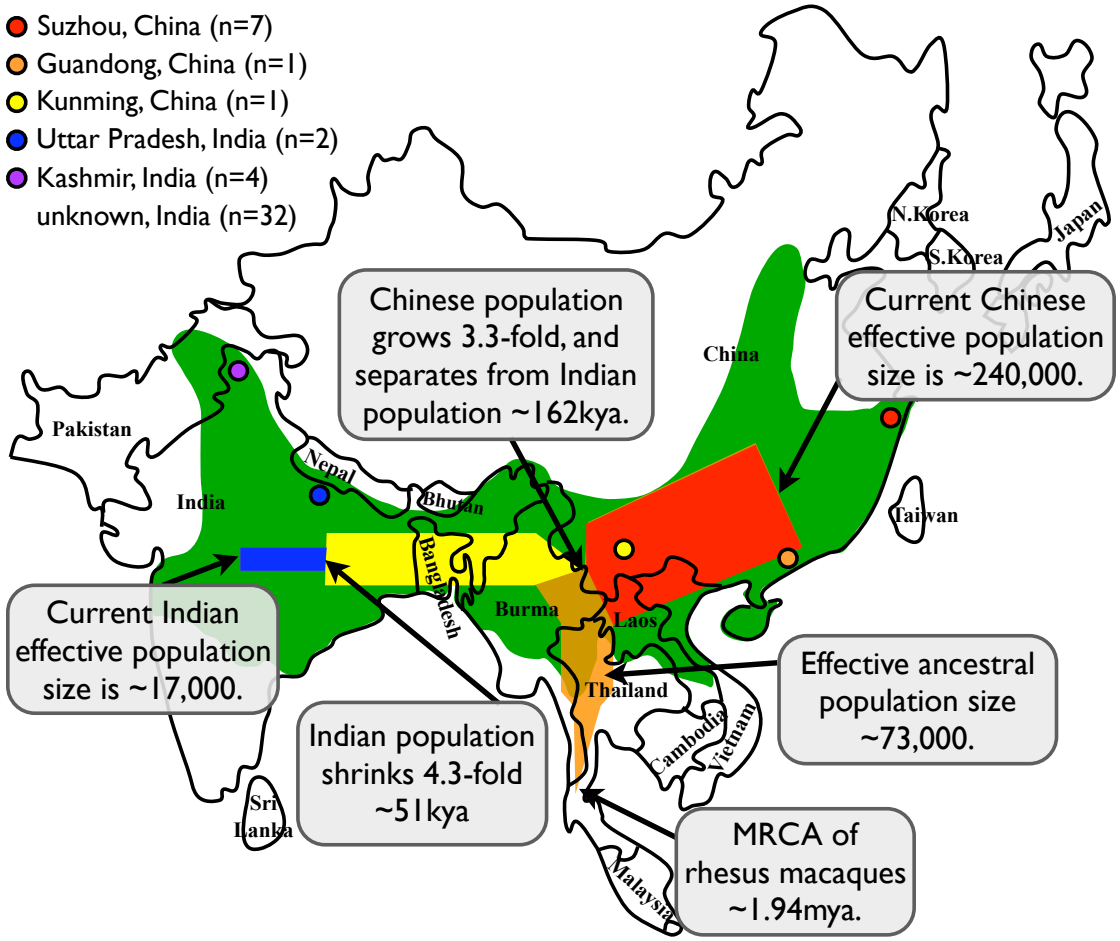

Figure 3.1: The current geographic range of rhesus macaques [green, redrawn from Fooden (1980)] with the inferred demographic history and the sample locations superimposed. The geographic location of the MRCA is based on Delson (1980).

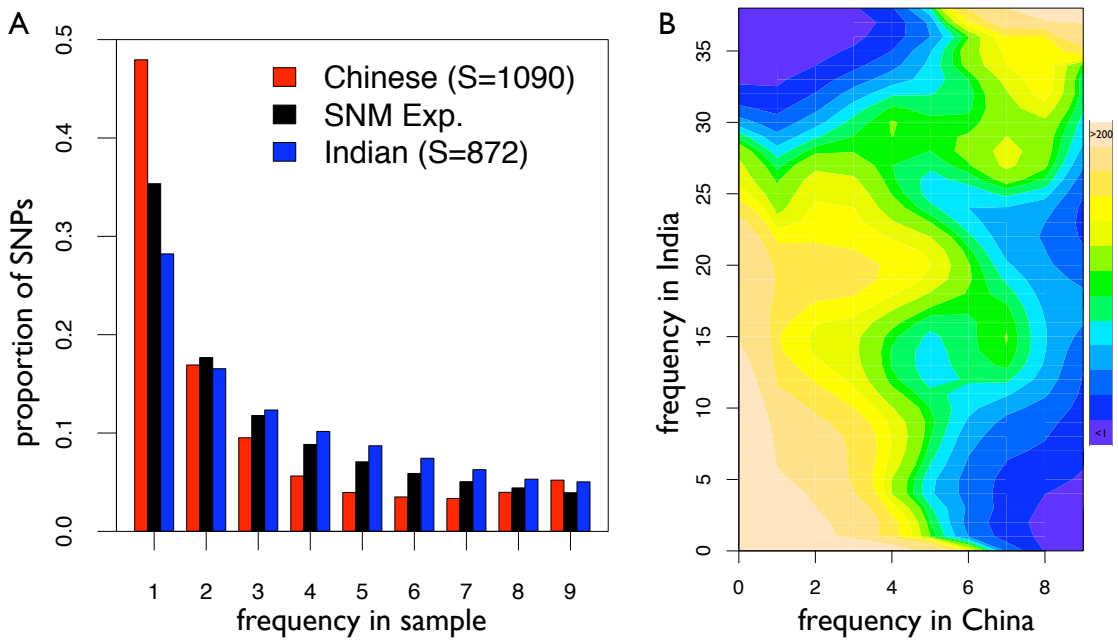

Figure 3.2: (A) The marginal frequency spectrum of derived mutations for each population (shown as expected proportions in a subsample of 10 chromosomes by integrating over possible configurations of observed and missing data, with the total number of SNPs in parentheses) and the expected distribution under the standard neutral model (SNM) of constant size. (B) A topographical map of the joint site-frequency spectrum for the two populations, with darker tones representing frequency pairs with few SNPs, and lighter tones representing frequency pairs with many SNPs.

nese population harbors an excess of rare SNPs relative to a population of constant size, whereas the Indian population has too few rare and too many intermediate- and high-frequency derived SNPs (Figure 3.2A). The observed disparity in SNP density (7.25 SNPs per kb for Chinese versus 5.8 SNPs per kb for Indian) in the two populations suggests that the effective size of the Chinese population is much larger than the Indian population, given that the Indian sample size is four times as large as that of the Chinese.

We observed a moderate level of population structure between the Indian and Chinese samples, as measured by Wrights F_{ST} statistic (average F_{ST} = 0.14; SD = 0.11; range = -0.024 to 0.645; Figure 3.3A). Furthermore, the Bayesian clus-

tering program STRUCTURE Falush et al. (2003) clearly separates Chinese and Indian individuals when assuming two clusters (Figure 3.3B), and considering more clusters does not significantly improve the fit of the model. We found only one Chinese individual with a marginal amount of Indian ancestry (8.5%, sampled from Suzhou), and eight Indian individuals with more than 5% Chinese ancestry (max 16.8%, including animals from all three primate centers; see Appendix A). These low levels of admixture suggest that recurrent migration between the populations has been minimal. Moreover, the two populations were clearly distinguished by principal components analysis (Price et al., 2006) along the first two axes of variation (Figure 3.3C). Interestingly, the second component also separates one Chinese individual (sampled from Suzhou) from the others, which suggests that further population substructure may exist. Although this individual is not differentiated from other Chinese-origin animals in the STRUCTURE analysis, it may, nonetheless, harbor alleles from an unsampled Chinese subpopulation (i.e., the two wild-caught parents may be from different subpopulations).

Using maximum likelihood under the assumption that the animals in this study form a random sample from their respective population (see Appendix A), we fit a two-population demographic model to the joint distribution of SNP frequencies, or site-frequency spectrum, shown in Figure 3.2B. Our model suggests that the Chinese population expanded by a factor of 3.3 and separated from the Indian population approximately ∼162 thousand years ago (ka) (95% confidence interval, CI = 183 to 132 ka)]. After separating, the Indian population maintained its ancestral population size until ∼51 ka (CI = 72 to 21 ka), when it was reduced by a factor of 4.3. The population genetic model, while a very simplistic approximation to the rich and complex history of the species, fits the data well, as indicated by a goodness-of-fit test (P=0.133). Coalescent simulations (see Appendix A)

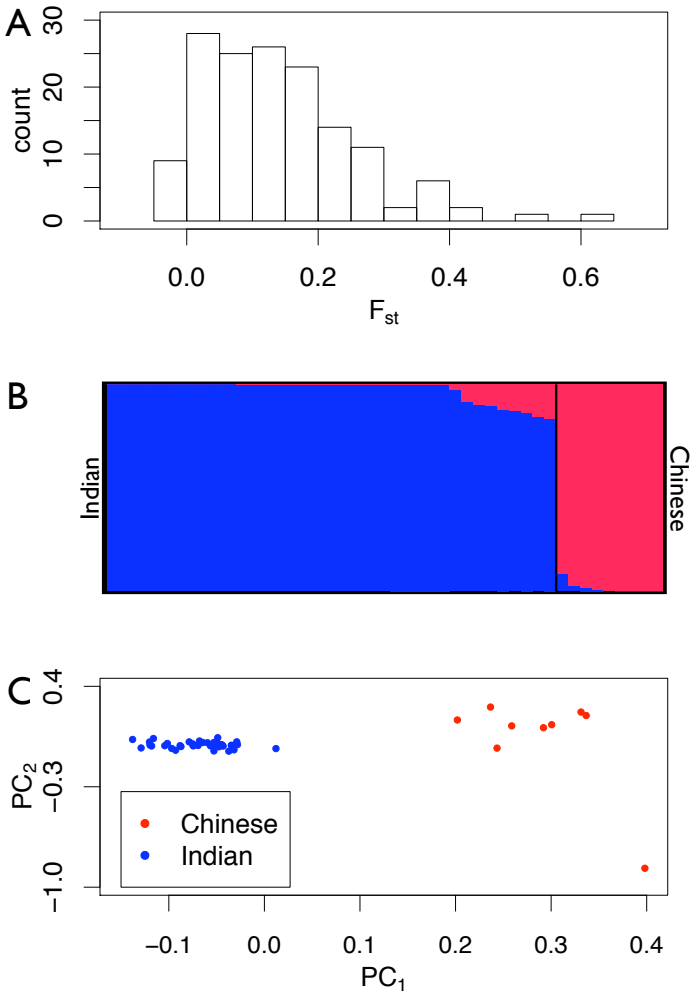

Figure 3.3: (A) The distribution of F_{ST} between Indian and Chinese rhesus, calculated with the average pairwise-difference across each nonoverlapping window (see Appendix A). (B) STRUCTURE results. Individuals are represented by vertical lines, and sorted by their amount of Chinese ancestry (black vertical line separates animals with Indian and Chinese origins). Colors correspond to the proportion of an individuals ancestry attributable to a given population (blue, Indian; red, Chinese). (C) Principal component 1 (PC1) and PC2 separate Indian from Chinese individuals. PC2 also isolates a single Chinese individual [corresponding to an individual sampled from Suzhou and shown as the fourth individual from the right in (B)].

on the basis of the inferred demographic history for Indian and Chinese rhesus macaques suggest that the MRCA of the two populations lived ∼1.94 Ma (SE 14 Ky). This estimate places the MRCA of rhesus near the divergence time from M. fascicularis, inferred from mitochondrial DNA to be 1.83 to 5 Ma (Hayasaka et al., 1996; Morales and Melnick, 1998). Moreover, our simulations suggest that the effective size of the ancestral population of rhesus macaques was ∼73,070 (SE 231) individuals, implying that the current effective size of the Chinese population is ∼239,704 whereas the Indian population is estimated to be ∼17,014.

The recent demographic events that caused these differences in effective population sizes of Indian and Chinese rhesus macaques have also had a large impact on linkage disequilibrium (LD). To quantify the extent of LD in Indian and Chinese rhesus macaques, we measured the correlation coefficient (r^2) of alleles from frequency-matched SNPs [see Appendix A and Eberle et al. (2006)]. Figure 3.4 shows substantial differences between the Indian and Chinese rhesus macaque populations, which are more extreme than the patterns observed among humans. For example, within the Indian rhesus population, LD extends much further than LD observed for European humans, whereas the Chinese rhesus population shows little LD, even for SNPs that are physically very close. Coalescent simulations (see Appendix A) show that the observed patterns of LD are consistent with our inferred demographic history of this species (shown in Figure 3.4 as light blue and pink curves for Indian and Chinese rhesus, respectively). However, LD in the Indian population extends slightly further than expected. This observation may be consistent with recent admixture with a Burmese rhesus population not sampled in this study (Smith and McDonough, 2005), since admixture between populations with allele frequency differences is known to generate long-range LD.

3.4 Discussion

In this study, we analyzed noncoding data in rhesus macaques to characterize their underlying demographic history, and to quantify the extent of LD relative to humans. The genetic differences that we have observed between Indian and Chinese rhesus macaques are consistent with a recent report on the distribution of SNPs in these populations (Ferguson et al., 2007), as well as previous studies of protein coding, microsatellite STR (short tandom repeat), MHC loci, mitochondrial and

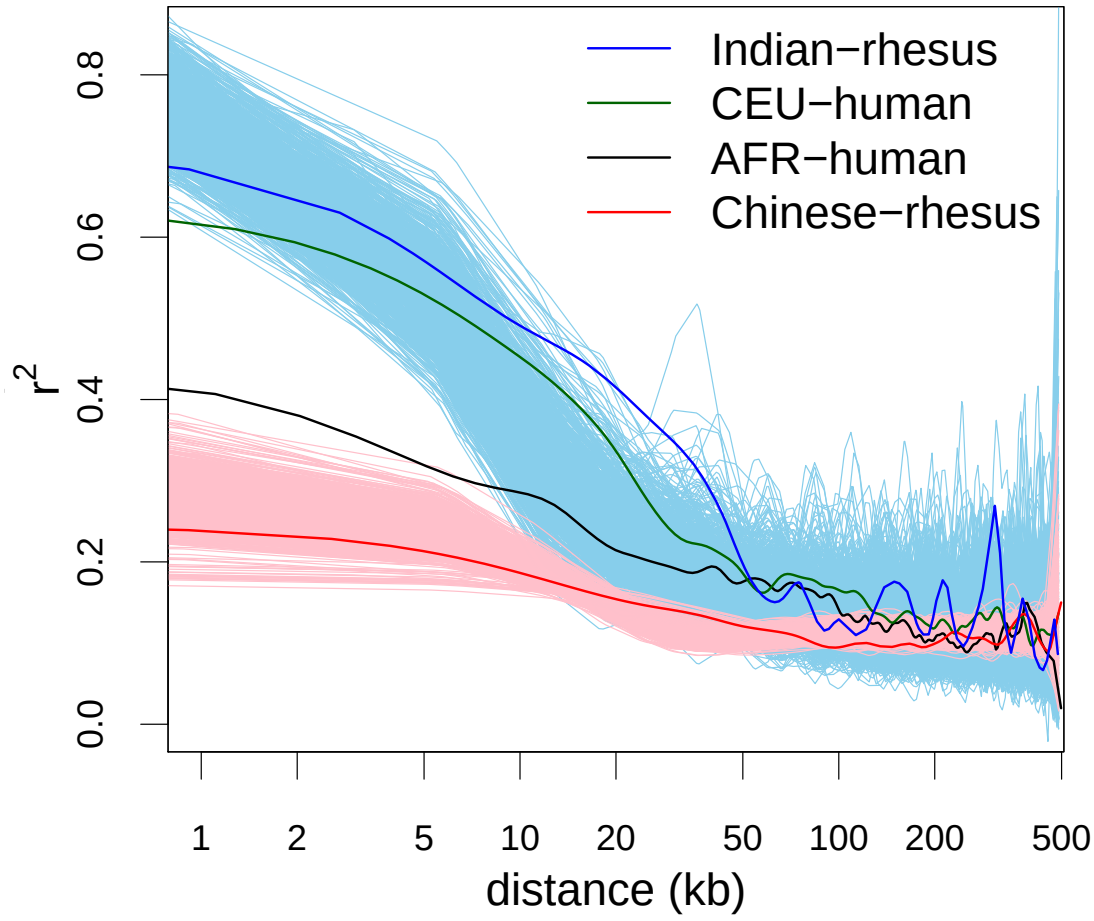

Figure 3.4: The decay of LD for Indian and Chinese rhesus macaques versus European and African humans (n = 9 for all samples), along with the decay of LD for 1000 neutral simulations of our inferred demographic history for rhesus macaque. Human data are from three ENCODE regions orthologous to the rhesus data [see Appendix A and HapMap (2005)].

Y-chromosome DNA haplotypes (Smith and McDonough, 2005). Without samples from wild caught Indian rhesus monkeys, however, these data must be regarded as estimates, because they may reflect a sampling bias toward those macaques that are available for study in the United States as a result of international restrictions on exportation of primates.

Extending these studies to whole-genome association mapping in captive-born animals could be fruitful for identifying genes involved in human diseases. Based on the patterns of LD that we have observed, such an association study would likely require many fewer markers to identify common disease-causing variants in rhesus macaques than in humans. Because LD in captive Indian rhesus macaque populations extends much further than in humans, a SNP map with roughly 1 SNP every 35 kb (82,000 SNPs total) would suffice to achieve the same threshold ($r^2 = 0.4$) as a marker every 6 kb in humans [see Appendix A and Kruglyak (1999)]. Furthermore, since LD decays much faster in Chinese rhesus monkeys than in humans, they provide an ideal platform for localizing mutations that are hard to map in either Indian macaques or humans as a result of extensive LD among candidate mutations in a particular region.

3.5 Acknowledgments

We thank the Yerkes, Oregon, and California National Primate Research Centers for contributing samples, and D. G. Torgerson for comments. Funded by NIH grant RR05090 to DGS, NIH RR00163 to BF, NIH RR015383 to J.R., NSF0516310 to CDB, and R01HG003229 to CDB, RN, A. G. Clark, and T. Matise.

CHAPTER 4

SELECTION ON FINITE SITES UNDER COMPLEX DEMOGRAPHIC EVENTS*

*In preparation for submission as an Applications Note to *Bioinformatics* with author Ryan D. Hernandez.

4.1 Abstract

Summary: We present a new forward population genetic simulation program that simulates the evolution several populations under a general Wright-Fisher island model. This program is highly flexible, allowing the user to simulate several loci with or without linkage, where each locus can be annotated as either coding or non-coding, sex-linked or autosomal, selected or neutral.

Availability: The source code (written the `C` programming language) is available at http://bustamantelab.cb.bscb.cornell.edu/software.shtml, and our web server (http://cbsuapps.tc.cornell.edu/sfscode.aspx) will allow the user to perform simulations using the high performance computing cluster of the Computational Biology Service Unit at Cornell University.

Contact: rh79@cornell.edu

4.2 Introduction

Forward population genetic simulations have long played a crucial role in evolutionary biology, and have been advocated nearly as long as computers have been available (Fraser, 1957). Simulations have been useful for guiding our intuition, testing theoretical approximations, and evaluating the power of statistical tests, yet they remain an underutilized tool in current research. By following an *in silico* population generation by generation and mimicking all stages of the life cycle, it is possible to simulate data under highly complex scenarios that capture many of the factors that affect natural populations. However, with complexity generally comes a computational burden, which has driven many studies toward simplified approximations.

In stark contrast to forward simulations, generating samples under the coalescent process (Kingman, 1982; Hudson, 1983a,b) can be extremely fast. Coalescent simulations start with a sample of chromosomes from the present and generate the history of the sample back to their most recent common ancestor (gaining speed by only keeping track of the evolutionary events and genetic material that directly contribute to the sample of chromosomes). However, when considering the effect of natural selection (particularly across many linked sites), one has little option other than forward simulations. Previous implementations of forward simulation programs [*e.g.*, Balloux and Goudet (2002); Dudek et al. (2006); Guillaume and Rougemont (2006); Hey (2004); Hoggart et al. (2007); Peng and Kimmel (2005); Sanford et al. (2007)] have produced a wide range of options geared toward mimicking natural populations. `SFS_CODE` adds flexibility to many of these options, and adds several new features.

Among the features implemented in `SFS_CODE` is an ability to simulate genes, whereby several adjacent loci can be annotated as either coding or non-coding (*e.g.*, exons and introns or up-/downstream regions). More generally, loci can be arbitrarily spaced (from physically adjacent to completely independent). In coding regions, new mutations are either synonymous or nonsynonymous (owing to the universal genetic code). Each locus can evolve neutrally or subject to natural selection (in coding regions, only nonsynonymous sites are subject to natural selection, while all non-coding mutations are potentially driven by selection). Selective effects can be drawn from a wide range of possibilities, including a mixture of Gamma distributions (positive and negative with user-defined mixture coefficients and parameters). Moreover, several mutation models have been implemented, from standard models of equal mutation rates (Jukes and Cantor, 1969) and transition-transversion biases (Kimura, 1980), to fully context-dependent models of mam-

malian evolution including CpG effects (Hwang and Green, 2004). By modeling each locus as having only finitely-many sites that can receive multiple mutations (but storing all mutations contributing to the final sample) more realistic data can be generated to better understand the factors contributing to observed sequence data.

Several populations can be simulated in a generalized island model of arbitrary migration rates to and from each population (which can also vary over time). Because both male and female sexes are maintained, it is possible to have biased sex ratios in each population and to allow sex-biased migration. Additionally, each population can experience its own demographic history, be exposed to differential effects of natural selection, evolve with different generation times, as well as go extinct at any time (*e.g.*, for the comparison of Neanderthal and human).

4.3 Materials and Methods

Most populations harbor individuals that are nearly identical at the DNA level. One can take advantage of this in order to construct an efficient simulation program. However, having the full DNA sequence is also important. `SFS_CODE` can simulate coding regions (*e.g.*, synonymous and nonsynonymous mutations in a codon structure) and implements a context-dependent mutation model (whereby the the mutation rate at each DNA site is dependent on both adjacent sites). In order to accomplish this, a single representative DNA sequence is stored in memory for each population (and updated after each fixation event). Additionally, since the number of haplotypes that are segregating in a population is usually much smaller than its effective size, only a single copy of unique haplotypes are stored in memory (with all individuals carrying identical haplotypes pointing to the same space in memory).

Table 4.1: Information returned for each mutation

data type	description
char	ancestral nucleotide
char	derived nucleotide
char	nonsynonymous (0/1)
char	sex-linked
char	5' nucleotide
char	3' nucleotide
int	ancestral amino acid
int	derived amino acid
long	frequency
long	generation arose
long	generation fixed
long	site
double	fitness

In `SFS_CODE`, a haplotype is stored as a binary tree, where each node represents a mutation carried by the haplotype. In order to make the program as broadly applicable as possible, many details regarding the mutation event are stored (Table 4.1). To maintain a balanced binary tree, haplotypes are implemented as Splay trees (Sleator and Tarjan, 1985), whereby a mutation will be brought to the top of the tree any time its contents are accessed (*e.g.*, whenever a recombination event occurs between two mutations).

At the beginning of the simulation, there is a single DNA sequence that is carried by every individual in the population. This sequence is drawn from the stationary distribution given by the mutation model, and requires a burn-in period of many generations to introduce new mutations and to reach mutation/selection balance (typically $5\times PN$ generations will suffice for a population with PN chromosomes). Upon the completion of the burn-in period, speciation events and demographic effects can occur. At the end of the simulation, a random sample of individuals (including all of their chromosomes) will be obtained without replacement.

The basic life cycle that takes place every generation is as follows. Each individual in the population is generated by picking both a male and female parent with probability given by their relative fitness in their sex (only necessary for diploid and tetraploid populations, haploid populations are simulated asexually). The new individuals are then able to migrate among populations. The gametes of the new population then undergo recombination and mutation before being passed on to the next generation.

4.4 Conclusions

Forward population genetic simulations are an extremely useful tool. We have developed a flexible program (`SFS_CODE`) that allows the user to simulate data under a wide range of scenarios. We have also developed a web browser for our simulation program to give everyone access to a high performance computing cluster.

By implementing a finite-sites mutation model, simulations can be performed to test the effect parsimoniously assuming that all mutations have been observed. However, by reporting all mutations contributing to sampled chromosomes, `SFS_CODE` can also be compared with infinite-sites models. Moreover, by reporting extensive details for every mutation, it is possible to extract the information needed for most population genetic questions.

APPENDIX A

SUPPLEMENTAL INFORMATION FOR CHAPTER 3

A.1 Data Collection

Animals Surveyed: Forty-seven rhesus macaques were sampled (9 Chinese and 38 Indian, Table A.1). Seven of the Chinese animals were sampled from Suzhou (eastern China, including the outlier seen in Figure 3.3B), one from Kunming (western China), and one from Guandong (eastern China). All of the Chinese-origin animals are captive-born from wild-caught parents. Because of the 1978 India export ban of rhesus macaques, all Indian rhesus samples were obtained from U.S., captive born animals. Though most of the U.S animals cannot be traced back to a specific location in India, four animals are known to be derive from Kashmir in northwestern India, and two animals are known to derive from Uttar Pradesh in northeastern India (see Table A.1; Figure 3.1). To minimize the possibility of including closely related rhesus in our analysis, we selected animals that are unrelated for at least 2-3 generations (no grandparents or great-grandparents in common) and, when possible, date back to the founding of their respective colonies.

DNA was sequenced across 5 ENCODE regions in all animals. ENCODE regions were chosen because they have been widely studied in dozens of other mammalian species that have not yet had their entire genome sequenced (in particular baboon, which shares a common ancestor with rhesus macaque ∼6-9 Ma). Within each ENCODE region, several small windows (mean length = 900bp) were chosen ∼5 kb apart, followed by several more windows spaced ∼50kb apart until the entire 500kb ENCODE region had been spanned. In the end, 166 non-overlapping windows were sequenced in each individual (150,376bp) to represent ∼2.5Mb (∼0.1%) of the genome. Our sequencing effort resulted in the discovery of 1476 SNPs. For Figure 3.3A, F_{ST} was calculated for each non-overlapping window using the program SITES, developed by J. Hey (`http://lifesci.rutgers.edu/~heylab/HeylabSoftware.htm`). For Fig-

ure 3.4, median r^2 values from 100 equally weighted bins were generated using Haploview (Barrett et al., 2005), with the human data obtained from the HapMap-ENCODE resequencing project (HapMap, 2005).

We identified the putative ancestral allele at each SNP by aligning the reference sequence for each window to the baboon clone corresponding to each ENCODE region using Blat v33.237 (Kent, 2002). Baboon clones NT_107571, NT_108362.1, NT_107340.3, NT_107966.1, and NT_107712.1 were downloaded from Genbank, and correspond to ENCODE regions ENm003, ENm010, ENr112, ENr233, and ENr321, respectively. We could not identify a putative ancestral state for 67 SNPs because there was either no orthologous baboon sequence (38 SNPs) or because neither of the segregating alleles matched the baboon sequence (29 SNPs).

Table A.1: Rhesus macaque sampling Locations.

ID	Primate facility[a]	Country of ancestry	locality
23502	ONPRC	China	Guandong
31438	CNPRC	China	Kunming
34150	CNPRC	China	Suzhou
34458	CNPRC	China	Suzhou
34472	CNPRC	China	Suzhou
34492	CNPRC	China	Suzhou
34493	CNPRC	China	Suzhou
34496	CNPRC	China	Suzhou
34576	CNPRC	China	Suzhou
17645	ONPRC	India	Unknown
17700	ONPRC	India	Unknown
18263	Yerkes	India	Unknown
18273	Yerkes	India	Unknown
19150	ONPRC	India	Unknown
19296	ONPRC	India	Unknown
24118	CNPRC	India	Unknown
24162	CNPRC	India	Unknown
24888	CNPRC	India	Kashmir
25442	Yerkes	India	Unknown
25505	Yerkes	India	Unknown
25704	Yerkes	India	Unknown
25712	Yerkes	India	Unknown
25750	Yerkes	India	Unknown
25751	Yerkes	India	Unknown
25754	Yerkes	India	Unknown
25755	Yerkes	India	Unknown
25934	Yerkes	India	Unknown
25935	Yerkes	India	Unknown
25937	Yerkes	India	Unknown
25946	Yerkes	India	Unknown
25952	Yerkes	India	Unknown
25959	Yerkes	India	Unknown
25962	Yerkes	India	Unknown
25973	Yerkes	India	Unknown
25974	Yerkes	India	Unknown
25983	Yerkes	India	Unknown
26301	CNPRC	India	Kashmir
26869	CNPRC	India	Unknown

[a]CNPRC, ONPRC, and Yerkes denote animals contributed from the California, Oregon, and Yerkes National Primate Research Centers.

A.2 Estimation of Demographic Parameters Using the Joint Site-Frequency Spectrum

A.2.1 Deriving the Model

For a sample of n_1 chromosomes from one population and n_2 chromosomes from the other, the joint site-frequency spectrum (JSFS) is a random matrix representing the number of SNPs that are observed at a frequency i out of n_1 in the first population, and j out of n_2 in the second [$i = 0, 1, 2, \ldots, n_1$; $j = 0, 1, 2, \ldots, n_2$; excluding the invariant cases ($i = 0$, $j = 0$) and ($i = n1$, $j = n2$)]. In our simple model, we assume that a panmictic ancestral population of size N_A split into two independent populations at some time t_1 in the past (where time in this study is scaled by $4N_A$ generations). One of the daughter populations represents the Chinese population, which instantaneously changed size to $f_1 N_A$ at the time of the split. The other represents the Indian population, which maintained the ancestral population size until some time t_2 in the past ($t_2 \leq t_1$), at which point the Indian population size instantaneously changed to $f_2 N_A$.

In this population genetic model, there are only four parameters to be optimized: $\theta = \{t_1, t_2, f_1, f_2\}$. The ancestral population size is a scaling factor, which can be inferred from these four parameters (with further assumptions regarding divergence and generation times, discussed below). Though our model does not explicitly allow for recurrent migration, we assume that each individual has some proportion of its ancestry from the opposite population (as inferred from our STRUCTURE analysis). Because the frequency of each SNP in each population is informative regarding the timing of demographic events, we have also incorporated the probability that the ancestral state of each SNP was misidentified (see discussion below).

We inferred the parameters of the demographic model (collectively denoted by θ) described above using the maximum likelihood approach of ref. Nielsen (2000) but modified to analyze data from more than one population. Let $p_{ij}(\theta)$ be the probability of observing a SNP at frequency i in population 1 (Chinese, say), and j in population 2 (Indian) under the demographic model θ, where $i = 0, 1, 2, \ldots, n_1$, and $j = 0, 1, 2, \ldots, n_2$ (with n_1 and n_2 the number of chromosomes sampled from population 1 and 2, respectively). The observed JSFS is then summarized by $X = (n_{1,0}, \ldots, n_{n_1,0}, n_{0,1}, \ldots, n_{n_1,1}, \ldots, n_{n_1-1,n_2})$, where $n_{i,j}$ is the number of SNPs where the derived allele is carried by i chromosomes in the first population and j chromosomes in the second (hereafter, we refer to such a SNP as being of size (i, j), and note that we only consider polymorphic sites). Assuming independence among SNPs, the likelihood function for the JSFS is then defined as

$$L(\theta) = \prod_{i=0}^{n_1} \prod_{j=0}^{n_2} \left(p_{i,j}(\theta)\right)^{n_{i,j}} . \tag{A.1}$$

This likelihood function assumes that all SNPs are independent. Should SNPs not be independent, it is then considered to be a composite likelihood function [which have been shown to result in consistent estimators of demographic parameters under a very general framework (Wiuf, 2006)].

To make inference using maximum likelihood, we calculate the expected frequency of a new mutation in terms of an expectation of coalescence times (Griffiths and Tavaré, 1999). Assuming the limit of small mutation rates, the probability of a SNP of size (i, j) is simply the ratio of the expected proportion of time on a coalescent tree during which a mutation could lead to a SNP of size (i, j). That is, for our set of demographic parameters θ,

$$p_{ij}(\theta) = \frac{E_\theta(t_{ij})}{E_\theta(T)} \tag{A.2}$$

Table A.2: Demographic Parameter Estimates.

Parameter	MLE[a]
t_1	0.0871 (0.0711, 0.0989)
t_2	0.0251 (0.01, 0.035)
f_1	3.2816 (2.333, 4.333)
f_2	0.2329 (0.133, 0.29)
$\mathrm{T}_{\mathrm{MRCA}}$	1.0219 (SE 0.0075)
T_{IN}	3.1338 (SE 0.0176)
T_{CH}	4.3111 (SE 0.0178)
T_{TOT}	5.6620 (SE 0.0180)

[a]Parentheses contain 95% CI unless indicated as SE

where t_{ij} is the sum of all branch-lengths in a coalescent tree on which a single mutation would lead to a SNP of size (i, j), and T is the total tree length (Nielsen, 2000). Both the numerator and denominator of equation A.2 can be approximated using standard coalescent simulations (Hudson, 2002) by simulating B coalescent genealogies under θ (in practice, we set B=2×10^6). For each genealogy, k, the total tree length (T_k) and sum of branch-lengths (t_{ij}) on which a single mutation could lead to a SNP of size (i, j) are calculated. Then $p_{ij}(\theta)$ is approximated by

$$p_{ij}(\theta) \approx \sum_{k=1}^{B} t_{ij} \left(\sum_{k=1}^{B} T_k \right)^{-1} \tag{A.3}$$

The maximum likelihood estimate (or composite maximum likelihood estimate, in the case of non-independence) of the demographic parameters are then the values of θ that maximize $L(\theta)$.

We optimized the likelihood by successively evaluating it on a grid of parameters, zooming in on the current approximate maximum likelihood estimate each iteration. Because of simulation variance, simplex bracketing and other optimizations based on derivatives of the likelihood function were not applicable. Maximum likelihood estimates (MLEs) of the demographic parameters are given in Table A.2.

Missing data were accounted for by summing over all possible states of the missing data [as in Clark et al. (2005)]. Because our inference is based on knowing the frequency of the derived allele at each SNP, knowledge of the ancestral state is required. We inferred the putative ancestral state under a parsimony assumption from alignments with baboon, but a parsimony assumption can lead to an excess of high-frequency derived mutations [mistaking the ancestral state of a low-frequency variant will make it appear as a high-frequency variant; Hernandez et al. (2007c,b)]. We therefore calculated the probability of misidentifying the ancestral state of each SNP assuming a context-dependent mutation model (as described in Hernandez et al. (2007b) using parameter estimates from Hwang and Green (2004) along the human lineage, as we were not able to obtain parameter estimates from old-world monkeys). To test whether the substitution parameters inferred along the human lineage are representative of the mutation patterns along the rhesus macaque lineage, we performed a linear regression of the context-dependent SNP classes (*i.e.*, the set of all mutations from trinucleotides $XYZ \rightarrow XWZ$, where W, X, Y, Z are any of the four nucleotides) in human and rhesus macaque. We found that there is a highly significant correlation in the two lineages ($r^2 = 0.69$, $p < 2.2 \times 10^{-16}$), suggesting that the underlying mutational patterns may be shared across primates.

Confidence intervals (CI) were generated by the likelihood profile method. For all values of a given parameter on a grid, the likelihood was maximized over all other parameters. This essentially results in a one-dimensional likelihood surface. Assuming a χ^2 approximation, an approximate 95% CI corresponds to all parameter values whose profile likelihood is within $\frac{1}{2}\chi^2_1(0.05) \approx 1.92$ of the maximum likelihood. It should be noted that if SNPs are highly correlated, these confidence intervals might be less accurate.

To perform the goodness-of-fit (GOF) test, we calculated a sum-of-squares

statistic (SSS) of the observed JSFS based on an expected JSFS generated from 5×10^6 simulations under the MLEs (using a reduced sample size of 10×10). We then simulated 1000 replicate datasets conditional on the total number of SNPs observed in each amplicon (assuming no recombination within an amplicon and complete independence across amplicons). For each replicate dataset, we calculated the SSS, and report the proportion of simulations with an SSS larger than our observed dataset.

The objective of the GOF test is to assess whether the observed data differ significantly from the predictions of the best fitting demographic model. In order to address this question, we estimate the parameters of the demographic model using composite likelihood and simulate data conditional on the MLE. For each replicate data set, we calculate the SSS, and assess significance of the observed SSS in light of these simulated SSS values (i.e., our p-values are calibrated to the observed variability of the SSS in the simulated data). Our simulations assume independence among amplicons and complete linkage within amplicons, which will give a larger variance in the GOF statistic than the assumption of complete independence among SNPs. We have conducted a sensitivity analysis and as long as the within-amplicon population recombination rate is below 0.001 per bp, our GOF p-value is above 5%, indicating a good fit of the observed data to the predictions of the model. This is equivalent to assuming that the recombination rate is below $\sim$10 cM/Mb, which is quite reasonable since human recombination rates are on the order of $\sim$1cM/Mb.

Another approach to inferring the parameters of the demographic model used in this study is to use the method IM (Hey, 2005; Nielsen and Wakeley, 2001). However, one of our co-authors is a co-developer of IM, R. Nielsen, and we have found that this method is not applicable to large-scale genomic datasets of the

type analyzed in this study.

A.2.2 Inferring the Most Recent Common Ancestor and Effective Population Sizes

Under the inferred demographic model, θ, coalescent genealogies can be simulated using standard software (Hudson, 2002). For each simulated genealogy, k, the time of the most recent common ancestor (TMRCA, denoted M_k) can be calculated by summing over $M^{(i)}$, the total time during which there are i lineages in the sample ($i = n_1 + n_2, n_1 + n_2 - 1, \ldots, 2$), and the total length of the tree can be calculated as $T_k = \sum_{i=2}^{n_1+n_2} iM^{(i)}$. The expected TMRCA is then calculated as the mean over B simulated genealogies (in practice we used $B = 5,000$).

To infer the effective size of the ancestral population, we used a method of moments estimator based on the expected number of segregating sites. From population genetic theory, the expected number of segregating sites in the absence of natural selection can be written as $E[S] = \theta TL$, where $\theta = 4N_A\mu$, μ is the mutation rate per site per generation, T is the expected length of the coalescent tree that relates all chromosomes sampled (with time scaled in terms of $4N_A$), and L is the number of nucleotide sites sequenced. We can then set the observed number of segregating sites equal to its expected value, and solve for

$$N_A = \frac{S}{4TL\mu}. \tag{A.4}$$

Assuming a generation time of 6.5 years for rhesus macaques and 6.6MY since its divergence with baboon (Steiper and Young, 2006), we can infer the substitution rate per generation from the alignments with baboon, resulting in μ=(1754substitutions) / ($2\times6.6\times10^6$years) / (145571bp) / (1gen./6.5years) =

$5.9 \times 10^{-}9$ substitutions/generation. From our simulations, T=5.66, so we then obtain N_A=(1476SNPs) / (150376bp) / (5.9×10^{-9}subst/gen) / 4 / 5.66 $\approx$ 73,070 individuals. Inferring the ancestral population size from the Chinese and Indian populations independently results in 70,863 and 78,062 (respectively, with T=4.31 for China and T=3.13 for India). The ancestral effective population size is a linear function of the generation time. Should the generation time be longer, the ancestral population size would be larger.

A.2.3 Converting Population-Scaled Times into Years

Time in a coalescent genealogy is scaled by the effective population size (in this case $4N_A$). A population-scaled time τ corresponds to $4N_A\tau$ generations, and by substituting equation A.4 for N_A, the generation time cancels. Therefore, the number of years, t, corresponding to τ can be found by:

$$t = \frac{\tau S}{LT\nu}$$

where S is the number of SNPs, L is the sequence length, T is the expected length of the tree, and ν is the substitution rate per *year*. Note that this value is linearly related to the divergence time between rhesus macaque and baboon. We have assumed that the two species diverged 6.6MYA [as was inferred by Steiper and Young (2006)]. If baboon and rhesus diverged much earlier than this (say 9MYA), then dates from our demographic inference should be multiplied by $\frac{9}{6.6}$.

A.2.4 Linkage Disequilibrium Simulations

We simulated patterns of linkage disequilibrium (LD) by generating 1,000 replicate datasets using the coalescent-based approach implemented in `ms` (Hudson, 2002).

Each dataset corresponds to 5 independent chromosomes of length ∼500kb. We assumed a constant mutation rate and recombination rate across the sequence (with the population-scaled recombination rate R set equal to the population-scaled mutation rate). The mutation rate was set such that the mean number of SNPs closely matched the observed number (Figure A.1). We dissected each simulated chromosome to extract 30 non-overlapping windows (with the first 15 windows spaced ∼5kb, followed by 15 more windows with spacing ∼50kb) to span the entire 500kb region. We then pooled the five chromosomes to obtain a single dataset, and treated it exactly the same way we treated the observed data. The `ms` command we used to simulate data was the following:

`./ms 94 5000 -t` θ `-r` R `500000 -I 2 18 76 -n 1 3.2816`
`-en 0.08714 1 1 -n 2 0.23286 -en 0.02591667 2 1 -ej 0.08714 2 1`

A.2.5 Number of SNPs required for genome-wide association study

Kruglyak (1999) suggested that a genome-wide association study would require a dense enough map to achieve $r^2 = 0.4$ between markers. In humans, it was suggested that this would require a common SNP every 6kb, or 500,000 markers distributed across the genome. Since LD decays much slower in Indian rhesus macaques than humans, achieving the threshold of $r^2 = 0.4$ would require a common SNP every 35kb. Since the rhesus genome is ∼2.87Gb (Rhesus Macaque Genome Sequencing and Analysis Consortium, 2007), this corresponds to a total of 82,000 SNPs.

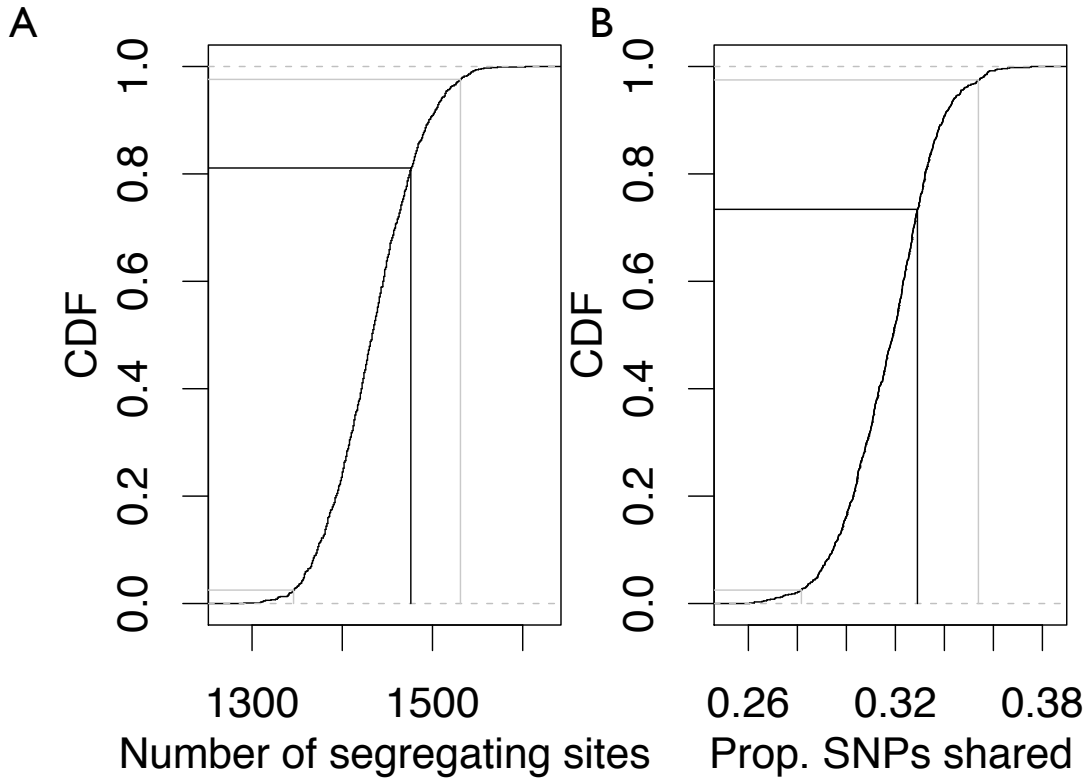

Figure A.1: Correspondence of our observed data with simulated data. The cumulative distribution function (CDF) for the number SNPs (A) and the proportion of SNPs that are shared between the populations (B). Gray lines indicate the 95% confidence interval, with the black lines showing our observed data.

APPENDIX B

USERS MANUAL FOR SFS_CODE

B.1 Preface

This is the user's guide to `SFS_CODE`. It outlines how to compile and use the program, but does not delve into many of the details regarding specific algorithms used or the underlying data structures implemented in the `C` source code. A subsequent verbose version (more of an *owner's manual*) of this document will be made available in the near future, and will delve into the gory details of how `SFS_CODE` works and is implemented. The verbose version will also include many examples and several unpublished simulation results that have been used to guide my intuition in population genetics, and will hopefully help you understand how to effectively use `SFS_CODE`. Table B.5 on page 124 outlines every option implemented in `SFS_CODE`, in includes a page reference indicating where each option was described in the text.

B.2 Overview

The program that this document is dedicated to can be described in a single run-on sentence as follows:

> SFS_CODE is a Wright-Fisher style forward population genetic simulation program for finite-site mutation models with selection, recombination, and demography.

This means that an entire population of individuals (and all their chromosomes) is followed generation by generation, from the beginning of the simulation to the time of sampling. This is contrary to coalescent simulations [such as ms; Hudson (2002)], where the history of a sample is simulated backward in time until its founder. SFS_CODE has the ability to simulate finite-site mutation models (meaning that some sites can receive several mutations). Nonetheless, SFS_CODE actually stores all mutations that are either segregating or fixed in at least one of the populations, so it can also act like an infinite-sites simulation program. However, its purpose is to generate a set of DNA sequences (an alignment) that can then be analyzed. This alignment, by the nature of the simulation, can therefore contain sites that have been the target of many mutations (as well as repeatedly being selected upon).

As described in further detail in subsequent sections, SFS_CODE allows the user to simulate highly detailed populations, with as much flexibility as ms. In addition to allowing for fairly complex demographic effects and migration schemes, SFS_CODE also allows the user to simulate coding versus non-coding regions, apply a distribution of selective effects to new mutations, generate domesticated populations, assume different male and female population sizes, linked and unlinked loci, sex and autosomal chromosomes, polyploids (haploid, diploid, or tetraploid), as well as a suite of built in or custom mutation models.

The basic algorithm used in this program is as follows:

1. Sample a sequence from the stationary distribution of the mutation model.
2. Burn-in a single population to mutation/selection balance.
3. Perform demographic and other evolutionary events.

4. Sample individuals from populations.

Each generation consists of the following components:

1. Produce each individual by randomly sampling a mother and a father from the previous generation (with replacement according to their relative fitness for their sex, unless simulating haploids, in which case there is no sex).
2. Randomly select individuals to migrate among populations.
3. Distribute a Poisson number of recombination/mutation events.

B.3 Getting Started

B.3.1 Compiling the Program

This section is only if you have downloaded the source code and wish to compile the program yourself. If you are using the web-based version of the program, then you can skip this section.

After obtaining and unzipping the distribution of this program, you will have a folder called `SFS_CODE`. Inside this folder, you will find (at least) two subdirectories `src` and `doc`. In the subdirectory `src`, you should find a `makefile`, along with several more subdirectories. The `makefile` will be used to compile all the programs provided with this distribution. It uses GNU's `gcc` compiler. Using your favorite command line terminal (Windows users should download and install `Cygwin` from `http://www.cygwin.com`), change directory to `SFS_CODE/src/`, and type `make`. This will create the directory `SFS_CODE/bin/`, which will contain the executables `sfs_code`, `convertSFS_CODE`, as well as any other programs in the current distribution.

If you get compiling errors, it is likely that either you do not have `gcc` installed, or the optimization flag `-fast` is not implemented for your operating system. If it is the former, then make sure you need to install `gcc`, or change the `makefile` to use your favorite compiler. If you are using `Cygwin`, you may need to update your version, making sure to install `gcc`. If it is the latter (or you get strange "Illegal instruction" errors at runtime), open the `makefile` using your favorite text editor (NOT Word, as you don't want to accidentally add any formatting flags to the file). Scroll down to about line 11, where it says `CFLAGS = ... -fast`. Replace the text "`-fast`" optimization flag with "`-O3`". Now proceed as before. If there are still problems, contact the author, and inform him of the system you are using. Note, if you are planning to use the Intel compiler, you may need to edit the source code `sfs_code.c` by uncommenting the very first line (this enables functions that Intel deems as "safe" but are not part of the standard `C` library, and will get rid of annoying warning messages).

B.3.2 Usage: Arguments at the Command Line

`SFS_CODE` is a command-line program. If you have already compiled the program, then you should be ready to go. Change directory to `SFS_CODE/bin`. A full list of options can be found in Table B.5 on page 124.

The basic command to run `SFS_CODE` is as follows:

```
sfs_code <Npops> <Niter> [<options> [arguments]]
```

Where `<Npops>` is the total number of populations you want to simulate, and `<Niter>` is the total number of iterations (or repetitions) you want to generate. In this documentation, arguments and options that are enclosed in <angled brackets> are required, and those in [square brackets] are optional. Subsequently, those in both angled and square brackets can be required in some potentially optional

instances (*e.g.*, `[<options>...]`, if you include anything after `<Niter>` then they must be options, which may contain required and/or optional arguments).

In `SFS_CODE`, all options have both a **long name** and a **short name**, except for timed events (beginning with '`-T`', described later, and only use the short name). For example, to set the mutation rate, you could use either "`-t` θ" or "`--theta` θ" to achieve the same result. Though both long and short names are case-sensitive, long names are of arbitrary length and tend to be more descriptive of the option. Short names are a single letter. Note that long names are preceded by two dashes ("`--`") while short names are preceded by only a single dash ("`-`"). Both the long and short names of all options are provided in Table B.5 on page 124.

In the text of this document, I will provide templates for each option, as well as numbered examples. In option templates, I will first give the long name, then the short name in parenthesis, followed by the format of its arguments, as in the following pattern:

```
--long_name (-short_name) [arguments]
```

As a first example, the help menu can be obtained using the option

```
--help (-h)
```

This means you would access the help menu by typing "`./sfs_code 1 1 -h`". In this special example, the number of populations and number of iterations do not need to be specified, so you could just type "`./sfs_code --help`" or "`./sfs_code -h`".

B.4 Running `SFS_CODE`

The most basic simulation is the following:

Ex. 1. `$ ./sfs_code 1 1`

Typing example 1 (excluding the `$`, which just represents the bash shell; in Windows, you also might not need the "`./`" bit either) into the command prompt will result in running a single iteration of the default simulation. The default parameter values are given in section B.10 toward the end of the documentation, and consists of simulating sequences of length 5000 nucleotide base pairs (5kb) from a "standard neutral" population of 500 diploid individuals, where the population scaled mutation rate $\theta = 0.001$/site with no recombination, from which a sample of 6 individuals will be drawn. By "standard neutral" population, I am referring to a population that is devoid of every evolutionary force other than mutation and drift. The full list of default parameter values is given in the Default Parameters section below.

The **mutation rate** per site ($\theta = 4N_e\mu$, for a diploid population) can easily be increased to a value of 0.01 per site using the option `--theta (-t) <`θ`>` as follows:

Ex. 2. `$ ./sfs_code 1 1 -t 0.01`

Recombination is just as easy to incorporate using the `--rho (-r) <`ρ`>` option, where $\rho = 4N_e r$ is the population scaled rate of recombination between adjacent sites for a diploid population. For example, the following would simulate a standard neutral population with per site mutation and recombination rates equal to 0.01.

Ex. 3. `$ ./sfs_code 1 1 -t 0.01 -r 0.01`

In general, you will want to do several (perhaps several thousand) simulations. Doing so requires some patience (this is a forward simulation, after all). However, **multiple simulations** can be performed at once by changing the parameter

<Niter>. Doing multiple simulations this way is beneficial, as compared to running them all independently, because SFS_CODE is able to take advantage of all the effort that went into all the previous burn-in periods. After an extensive initial burn-in period, the population will be at stationarity. It is much easier to obtain an independent draw from a population at stationarity than it is to reach stationarity. Figure B.1 shows how this is done.

The default initial burn-in time is $5\times PN$ generations, while subsequent burn-in periods are only $2\times PN$. You can change the initial burn-in time using

--BURN (-B) <burn>

and change the subsequent burn-in periods (for iterations > 1) using

--BURN2 (-b) <burn>

This would set the initial or subsequent burn-in times to <burn>$\times PN$ generations.

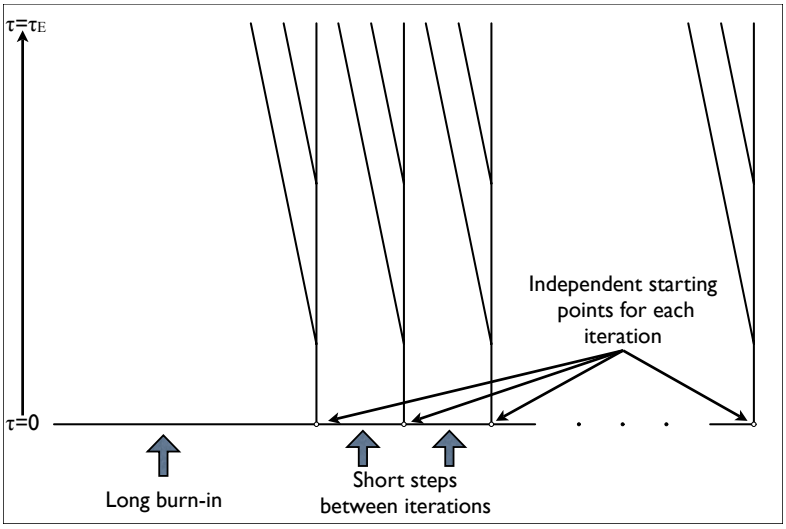

Figure B.1: Simulating multiple iterations in SFS_CODE begins with a long burn-in time, followed by relatively short steps ($\sim 2PN$ generations) between each iteration. Ancestral information at the beginning of each iteration is stored, such that the each starting point is a random draw of a population at mutation/selection/drift balance (each iteration uses the burn-in of all previous iterations).

In SFS_CODE, it is also possible to simulate an **arbitrary number of loci** (linked or unlinked) of arbitrary length using the following option.

`--length (-L) <nloci> <L1> [<L2>...<Lnloci>] [R]`

This option allows you to simulate `<nloci>`. The first locus will have length `<L1>`. You can stop here to set all loci to the same length. Otherwise, you have two options. You can specify each of `<L2>...<Lnloci>` to set the lengths of each locus, or if you have a repeating pattern (*e.g.*, a short locus followed by a long one) you can specify a subset of lengths followed by the character '`R`'. For example, if you want to simulate 4 loci, with lengths (500bp, 1kb, 500bp, 1kb), then you could use either of the following commands.

Ex. 4.
```
$ ./sfs_code 1 1 -L 4 500 1000 500 1000
$ ./sfs_code 1 1 -L 4 500 1000 R
```

You can **change the linkage among loci** using the next option.

`--linkage (-l) <p/g> <d1> [<d2>...<dnloci-1>] [R]`

The first argument to this option must either be '`p`' or '`g`', indicating whether the distance between loci will be be `<p>`hysical distance (in basepairs) or `<g>`enetic distance (recombination fraction). The second argument is the distance between the first two loci. This is all you need if you want all adjacent loci to have the same distance. Otherwise, (again) you have two options. You can either specify the distance between each pair of adjacent loci (*i.e.*, provide `<nloci>-1` values), or, if you have a repeating linkage structure you can specify a subset of distances followed by the character '`R`'. For **independent loci**, you can use "`--linkage p -1`" or "`--linkage g 0.5`". As an example, consider simulating 2 independent genes, each having 4 exons with lengths as in example 4 that are equally spaced with 2kb introns. You could simulate this as follows.

Ex. 5. `$ ./sfs_code 1 1 -L 8 500 1000 R -l p 2000 2000 2000 -1 R`

Moreover, you can **annotate** each locus as being either coding or non-coding, and sex or autosomal. By default all loci are autosomal coding regions. If you would like to specify whether each locus is **coding or not**, use the following option:

`--annotate (-a) <a1> [<a2>..<aR>] [R]`

where $\texttt{a}_i$ = '`C`' or '`N`' to indicate that the ith locus is coding or non-coding (respectively). If you want all loci to have have the same coding/non-coding annotation, just specify `<a1>`. Otherwise, you can either specify the annotation of all R loci, or specify the pattern to be repeated followed by the character '`R`'. To specify whether each locus is **sex or autosomal** (in an XX-XY sex determination system), use the following option:

`--sex (-x) <x1> [<x2>..<xR>] [R]`

which has the same structure as option `--annotate`, but $\texttt{x}_i$ = '`0`' or '`1`' to represent autosome or X-linked (respectively). Sex-linkage in a tetraploid population is a four-chromosome analog of the XX-XY sex determination system in diploids. In general, male sex chromosomes are simulated as if they were allopolyploids. For non-XX-XY mating systems, it may be possible to switch the meaning of male and female to achieve the desired effect, but must be done with caution.

Note that options `--linkage`, `--annotate`, and `--sex` must be specified **after** indicating the number of loci to simulate using option `-L`.

The **ancestral population size** used in a population genetic simulation is not as important as one might imagine (so long as all parameters are population-scaled, the actual size cancels). However, it can be changed from the default of 500 using the following option.

```
--popSize (-N) [P <pop>] <size>
```

This option would set the ancestral population size to the value `<size>`. For efficiency sake, the value you use should be kept as small as possible (but no smaller!!). The default is 500 diploid individuals, which should be sufficient for most purposes. However, if you are simulating a distribution of selective effects where the mean of the distribution is greater than the population size (in absolute value), then the entire population might go extinct. A realistic distribution inferred from human polymorphism data might induce such an effect.

B.4.1 Population Expansions and Bottlenecks

Natural populations fluctuate in population size, and any simulation program should accommodate this biological feature. However, it is often not necessary to simulate the exact trajectory of the population size, just the major trends (*i.e.*, the time of an expansion, or the severity of a contraction along with the degree of recovery). `SFS_CODE` implements four **demographic events**:

1. set the population size to a **new value**:

   `-TN <`τ`> [P <pop>] <`$N_{\texttt{new}}$`>`

2. change the population size by a **relative amount** ($\nu = N_{\text{new}}/N_{\text{old}}$):

   `-Td <`τ`> [P <pop>] <`ν`>`

3. allow the population size to start changing **exponentially**:

   `-Tg <`τ`> [P <pop>] <`α`>`

4. or commence **logistic** growth/decay:

`-Tk <`τ`> [P <pop>] <K> <r>`

Each of these options begin with '`-T`'. This indicates to `SFS_CODE` that an evolutionary event will occur at a specific time (`<`τ`>`, the first argument). The next character (one of '`N`', '`d`', '`g`', or '`k`') indicates the type of demographic event (NOTE: only short names are accepted for timed events). The first argument for these options is the time parameter `<`τ`>`. Time is scaled by the effective size of the *ancestral population* (essentially the number of generations *since the end of the burn-in* divided by the number of chromosomes in the ancestral population). Next there is an optional parameter that would allow you to specify a specific population. If you want the demographic event to be applied to all populations (or you are only simulating a single population), then this is not necessary. Otherwise, if you only want to apply the demographic effect to population 0 (see description below on how to simulate multiple populations), then you would use '`P 0`' here. Using the character '`P`' in your command tells `SFS_CODE` that the next parameter is a population and not the value for the size change effect.

Finally, if you are using '`-TN`' include the **new size** of the population `<`$\mathtt{N_{new}}$`>`. If you are using '`-Td`' include the **relative size** change `<`ν`>` = new size/current size (note that current size is NOT necessarily the ancestral size if you have multiple changes). If you are using '`-Tg`' include the **exponential** rate of growth/decay `<`α`>`. The parameter α determines the size of the population at time t by the equation $N(t) = N_0 e^{\alpha(t-\tau)}$, where time is scaled by PN_A (the number of chromosomes in the ancestral population, n.b. in a diploid population $P = 2$), τ is the time that the population size started changing, and N_0 is the size of the population when it started changing (not necessarily the ancestral size!). This implies that if you want the population to grow from N_0 individuals to N_F individuals in $(t-\tau)\times PN_A$ generations, you would invert the exponential equation to find $\alpha = \ln\left(\frac{N_F}{N_A}\right)/(t-\tau)$.

If you are using '`-Tk`' for **logistic** growth, include the carrying capacity `<K>` (the final population size) and the rate to approach it `<r>`. For logistic growth, the size of the population at time t is determined by the equation $N(t) = \frac{KN_A e^{r(t-\tau)}}{K+N_A(e^{r(t-\tau)}-1)}$.

`SFS_CODE` is a forward simulation program, so it thinks about time going forward. You can think of the burn-in period as "negative time", with the simulation actually starting at time zero (when the burn-in ends), and progressing forward in generations. Rather than referencing a specific number of generations, however, time is referenced in terms of PN_A generations, where N_A is the ancestral (original) population size and P is the **ploidy** (if you are simulating a diploid population, then $P = 2$ [the default], while $P = 1$ for a haploid population and $P = 4$ is a tetraploid population). You can change the ploidy using the following option:

```
--ploidy (-P) <P>
```

where `P` can be 1, 2, or 4. If `P`=4, you can specify either autotetraploid population or allotetraploid using

```
--tetraType (-p) <0/1>
```

where 0 indicates auto- and 1 indicates allotetraploid.

Keep in mind that the time scaling does not change as the population sizes change (though the amount of evolution taking place each generation can be considerably different). This is similar to `ms`, but instead of having a diploid time scaled in units of $4N_0$ generations (with N_0 the size at the time of sampling), `SFS_CODE` would scale time in units of $2N_A$ generations.

In `SFS_CODE`, it is also necessary to tell the simulation program **when to end** using the option

```
-TE <τ> [pop]
```

where again, time (τ) is scaled in units of PN_0 generations. In the simple applications above, the simulation actually ended when the burn-in period was over (*i.e.*, at time $\tau = 0$). In general, you can end the simulation for any population at any time (useful for generating samples from now extinct populations, such as neandertal), but in most situations you will terminate the evolution of all populations when you sample at the end of the simulation. To be more specific, the simulation ends when the last evolutionary event takes place. The "`-TE`" option just allows you to put a place holder until a specific generation.

If you want to simulate a model for an *African* population of humans, you might consider a simple 2-epoch model, where there was a constant ancestral population size (N_{A}) which instantaneously changed by a factor $\nu = N_{\mathrm{C}}/N_{\mathrm{A}}$ some time τ ago (in units of $2N_A$ generations). A diagram of this model is shown in Figure B.2. To implement this model in `SFS_CODE`, you would consider time during which the population has its ancestral size as the burn-in period. At the end of the burn-in period, the population instantaneously grows by a factor ν, and maintains the new size for $2N_{\mathrm{A}}\tau$ generations, when the simulation ends. Abstractly, this is implemented in `SFS_CODE` as

Ex. 6. `$ ./sfs_code 1 1 -Td 0` ν `-TE` τ

Notice that the demographic event actually occurs at time zero, with the population maintaining it's new size for τ units of time until the simulation ends. The parameters of such a model were inferred by Boyko et al. (2007) using synonymous SNPs across the human genome from an African American (AA) population. Their inferred demographic model is shown in Figure B.2. Simulating the AA demographic history using their inferred parameters is easy:

Ex. 7. `$ ./sfs_code 1 1 -Td 0 3.3 -TE 0.4377`

The equivalent command in `ms` would be:

```
ms 12 1 -t 16.5 -eN 0.066 0.303.
```

Note that `ms` requires $\theta = 16.5$. This ensures that the ancestral population has $\theta = 5$, which is the case for the `SFS_CODE` simulations (θ/per site = 0.001 across 5kb).

A simple demographic model for European populations is a 3-epoch bottleneck model. This model is also shown in Figure B.2, and consists of an ancestral population size (N_{A}), a bottlenecked population size (N_{B}), and a current population size (N_{C}). In `SFS_CODE`, generations begin accumulating when the first demographic event occurs (*i.e.*, $\tau = 0$, when the population decreases in size). The second demographic event occurs at the end of the bottleneck ($\tau^{2}_{\rightarrow} = 7703\mathrm{gen.}/(2N_{\mathrm{A}}) = 0.48$), and the simulation ends at $\tau^{\mathrm{E}}_{\rightarrow} = 8577\mathrm{gen.}/(2N_{\mathrm{A}}) = 0.54$. Given these parameters, this model is also straightforward to implement:

Ex. 8. `$ ./sfs_code 1 1 -Td 0 0.722 -Td 0.48 5.27 -TE 0.54`

The corresponding command in `ms` would be:

```
ms 12 1 -t 19.02 -eN 0.00728 0.19 -eN 0.0714 0.263.
```

You can **increase the sample size** using the option

`--sampSize (-n) [P <pop>] <SS`$_1$`> [<SS`$_2$`>...<SS`$_{\mathtt{Npops}}$`>]`

If you are only simulating a single population or you want to sample the same number of individuals from each population, then you can simply use "`-n <SS>`". If you want to set a specific sample size for each of n populations, use "`-n <SS`$_1$`>...<SS`$_n$`>`". Alternatively, if you just want to change the sample size of population i, then use "`-n P` i `<SS`$_i$`>`". Note that *individuals* are sampled, so if you simulate a diploid population (P=2), then 2 chromosomes will be printed at each locus for each individual.

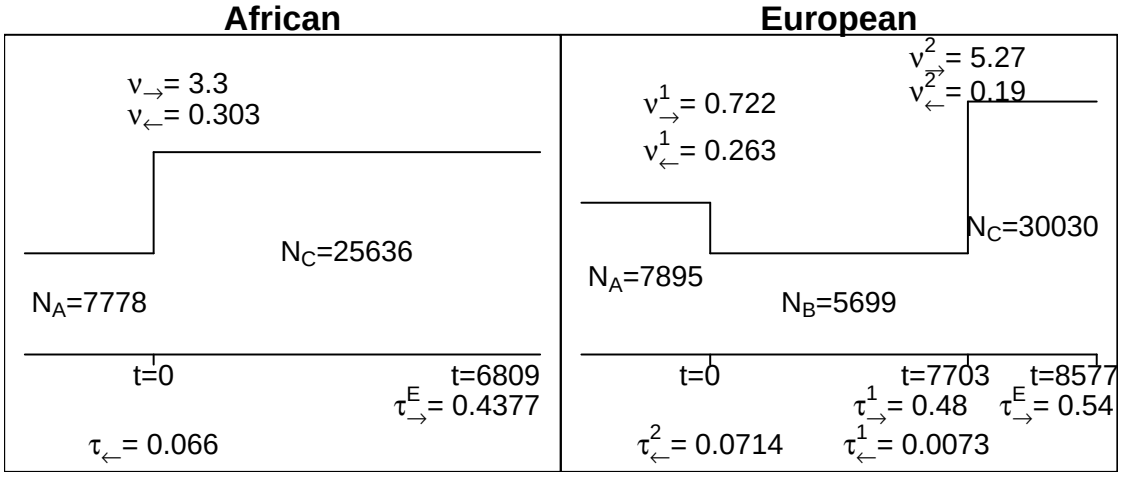

Figure B.2: The simple demographic scenarios considered considered in section B.4.1. Parameters (τ and ν) with subscript $\rightarrow$ are for `SFS_CODE` (forward time), while those with subscript $\leftarrow$ are for `ms` (pastward time). The horizontal axis represents time in generations (with $t = 0$ at the first demographic event). To obtain $\tau_{\rightarrow}$, divide the accumulated number of generations by $(2 \times N_A)$. To obtain $\nu_{\rightarrow}$ divide the new population size by the current population size at each transition. This methodology differs from `ms`, where the population size at time of sampling is generally the base. The number of generations and the effective population sizes for both populations were inferred by Boyko et al. (2007).

B.4.2 Distribution of Selective Effects

One of the many important components of a forward population genetic simulation program is natural selection. `SFS_CODE` assumes a simple multiplicative model of genic selection. This means that the fitness of an individual is just the product of the fitness effects of each mutation they carry. In general, a new mutation will have fitness $1 + s$, where s is the selective effect ($s > 0$ indicates positive selection, $s < 0$ indicates negative selection, and $s = 0$ indicates neutrality). An individual that is homozygous for such a mutation would then have fitness $(1+s)^2$. The selection coefficient is related to the population scaled selection coefficient $\gamma = 2N_e s$. Because population genetic theory is generally based on inference of γ, `SFS_CODE` draws γ from a specified distribution (discussed below), then divides it by PN_{C}, the number of chromosomes in the population when the mutation arises

(note that P is the ploidy, which is 2 for the default diploid population). `SFS_CODE` then uses s to determine the fitness of each individual, and normalizes by the mean fitness in the population.

It is important to note that `SFS_CODE` only implements *shift* models of selection. This means that as soon as a selected mutation is fixed in the population, the fitness effect of the site returns to 1. This avoids problems such as Muller's Ratchet, where the accumulation of deleterious mutations drives the population into the ground. Shift models are also in contrast to models such as the *House of Cards* model that was developed by T. Ohta in the 1960s (whereby assuming a normal distribution of selective effects will eventually lead to the fixation of an allele with selective effect ≥ 8 standard deviations above the mean, at which point evolution nearly halts).

You can **specify the distribution of selective effects** using the following option:

```
--selDistType (-W) [P <pop>] [L <locus>] <type> [args]
```

where `<type> [args]` are outlined in Table B.1 on page 92, and the optional flags '`P`' and '`L`' allow you to specify a single population or locus (respectively, if simulating more than one population or locus). For example, to simulate rampant positive selection, where all new nonsynonymous mutations have $\gamma = 5.0$, you would use

Ex. 9. `$ ./sfs_code 1 1 -W 1 5.0 1.0 0.0`

To simulate a situation in which 70% of new nonsynonymous mutations are deleterious with $\gamma = -5$, 10% are advantageous with $\gamma = 5$, and the remainder are neutral, you would use:

Ex. 10. `$ ./sfs_code 1 1 -W 1 5.0 0.1 0.7`

For a more complicated scenario, in which you want a distribution of positive and negative selection, we have `<type>=2`, which implements a mixture of Gamma distributions ($\Gamma(\cdot)$), one that corresponds to positive values of γ and one that has been reflected across the y-axis to capture a distribution of negative values. For example, if you want to assume that 90% of new nonsynonymous mutations are deleterious with a selection coefficient drawn from $\Gamma(1, 1)$ (a simple exponential distribution) and the remaining 10% are advantageous and drawn from $\Gamma(50, 10)$ (having mean = 5 and variance = 0.5), then you could use the following example.

Ex. 11. `$ ./sfs_code 1 1 -W 2 0.1 50.0 10.0 1.0 1.0`

Note that for example 11, the distribution of deleterious effects reduces to an

Table B.1: Selection: arguments for option `--selDistType` (`-W`)

<type>	[args]	description
0	$\emptyset$	**Neutral** (gamma = 0 for all mutations).
1	<GAMMA> <p_pos> <p_neg>	**3-point mass model**. Single γ (> 0) for both deleterious and advantageous mutations. With probability <p_pos> the sign is positive, with probability <p_neg> it is negative, otherwise with probability 1-<p_pos>-<p_neg>, $\gamma = 0$.
2	<p_pos> <aP> <lP> <aN> <lN>	**Gamma (Γ) distributions**. With probability <p_pos> $\gamma \sim \Gamma$(<aP>,<lP>) (mean = aP/lP, var. = aP/lP^2), otherwise $\gamma \sim$ -Γ(<aN>, <lN>).
3	<mean> <var>	**Normal distribution**. Mean = <mean> and variance = <var>.
4	$\emptyset$	**Advanced option**. Predefine distribution in file gencontextfreq.c, see text.

exponential distribution, while the distribution of advantageous effects has a mean and mode at 5. This mixture distribution is shown in Figure B.3. Of course if you simply want a Γ-distribution of negative selection (assuming no positive selection), then you can simply set `<p_pos>` = 0.

The fourth `<type>` (number 3), is a simple normal distribution. With a mean of zero, Cutler (2000) refers to the normal distribution of selective effects a model of positive selection. This is because on average, half of the new mutations will be advantageous, and a majority of the deleterious mutations will be eliminated.

The final `<type>` of model for the distribution of selective effects is an "advanced" option. For this option, you can create as complicated a distribution as you'd like in another statistical package (`R`, for example). This distribution can be discretized into 100 bins of equal density (using the `quantile` function in `R`, for example). These 100 bins are then copied into the vector `fitQuant` that is stored in the file `gencontextrate.c`. After changing this vector, the program must be recompiled (this is the only reason that it is referred to as an "advanced" option... more realistically, it is a rudimentary option that requires more work, but provides the ultimate flexibility). This model is actually preferred to `<type>=2`, as it is much quicker to randomly sample from a discretized distribution than it is to draw from a mixture of Γ-distributions. However, population size changes cannot be accommodated with this option.

It is also possible to specify that one population remain a neutral population. This can be useful if you want to specify a common distribution of selective effects for all populations but one. This is done using

`--neutPop (-w) <pop>`

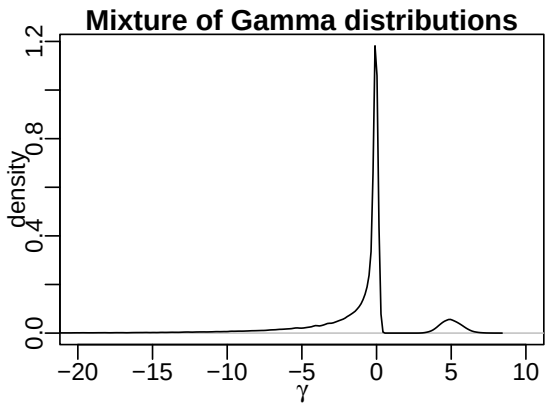

Figure B.3: A distribution of selection coefficients, where 90% of new mutations would be deleterious with $\gamma \sim -\Gamma(0.231, 0.1279)$, and the remaining are drawn from $\gamma \sim \Gamma(50, 10)$.

Selective Effects with Demography

When population sizes change, the relative effect of selection changes (selection is stronger in a larger population). This is accommodated by altering the distribution of the population scaled selection coefficient. For constant (type 1) and normal distributions (type 3) this is easily accommodated by adjusting the mean. For a mixture of Gamma distributions (type 2), the λ_N and λ_P parameters are adjusted such that the new means correspond to the change in population size. This also affects the variance, in accordance with the Gamma distribution. However, for the custom distribution of selection coefficients (type 4), population demography cannot be accommodated. If a custom distribution of selection coefficients is used and the population sizes change, then the same distribution of γ will be used (thereby inflating/deflating s to maintain a constant value of γ).

Selective Constraint

In the way that Kimura outlined the *neutral model of evolution*, some proportion of nonsynonymous mutations are completely lethal, and never contribute to polymorphism. All other nonsynonymous mutations were completely neutral, and had

no selective effect at all. As a result, it is often of interest to simulate data under such a neutral model (or allowing some proportion of nonsynonymous mutations to be lethal in general while the remaining nonsynonymous mutations follow the specified distribution of selective effects). This also generalizes to non-coding regions, where some proportion of mutations can be lethal. In Kimura's model, the parameter f_0 represents the proportion of neutral mutations. In `SFS_CODE`, you can adjust the **non-lethality parameter** using the following command.

```
--constraint (-c) [P <pop>] [L <locus>] <f0>
```

This option can even be used when simulating non-neutral models of evolution, as a way of signifying that only some mutations will contribute to polymorphism.

The way this option works, is that for each nonsynonymous or non-coding mutation, with probability $1 - f_0$, the fitness effect will be -1. This effectively sets the fitness of the individual to zero (as the fitness of the individual is defined as $1 + s$). This means that any mutations that are unique to this individual will also be lost in the next generation, as it will not pass on any of its gametes. All synonymous mutations are assumed to be neutral (*i.e.*, none are considered lethal).

B.4.3 Multiple Populations

In the above examples, we have used exclusively a single population, with `<Npops> = 1` as the first parameter into `SFS_CODE`. If we change this parameter, then we can simulate multiple populations. Note that populations are numbered from 0 through `<Npops>`-1.

There are two ways to create new populations. You can either have a speciation event or a domestication-style event. For a **speciation event**, one population will be split into two identical populations (equal size, etc.). To split population `i` into two populations (`i` and `j`) at time τ, you use the following template.

`-TS <τ> <i> <j>`

For a **domestication event**, one population (`i`) will be split into two (`i` and `j`), but the second population will primarily be composed of individuals that carry a particular derived allele, chosen at random from all the alleles that have a specified frequency (within 5% of `<allele_freq>`). After choosing a particular allele from the founding population, `SFS_CODE` will randomly sample individuals that are homozygous for the allele. If there are not enough homozygous individuals, then it will choose from the heterozygous individuals. If there are still insufficient individuals, then it will randomly choose non-carriers, until the specified population size, `<N>` is reached (note that `<N>` must be less than the size of the parent population `i`). The template for this option is as follows.

`-TD <τ> <i> <j> <allele_freq> <N> [locus]`

If a `locus` is specified, then `SFS_CODE` will try to find an allele in that particular locus (not necessary if only simulating a single locus). If `locus` is not specified, then `SFS_CODE` will start at the center-most locus that is simulated. If there isn't an allele near the specified `allele_freq`, `SFS_CODE` will search adjacent loci until one is found. Failing to find any mutations at the specified frequency, `SFS_CODE` will select the allele that is closest in frequency.

Now that multiple populations have been initialized, it is essential to tell `SFS_CODE` when to **end the simulation**. This was mentioned above with regards to demographic effects, but is worth mentioning again. This is done using the familiar option `-TE <τ> [pop]`. As an example, say you wanted to simulate human polymorphism data with a chimpanzee outgroup (assuming a population scaled divergence time of $\tau = 10$ and an allopatric speciation event). You could use the following:

Ex. 12. `$ ./sfs_code 2 1 -TS 0 0 1 -TE 10`

This example would first generate a single population at stationarity during the burn-in. At the end of the burn-in ($\tau = 0$), the population would be split into two identical populations, which would evolve independently until the end of the simulation ($\tau = 10$).

As an example of a domestication event, consider a model for dog breed formation, where you also want to simulate the ancestral dog population. This model is characterized by a major bottleneck in the ancestral population followed by rapid growth. Then, after growing for some time, 2 new breeds (of size 100 and 10) are formed using alleles at frequency 0.1 and 0.01 (respectively) in the ancestral population. These new breeds are then simulated for $0.1 \times 2 \times 500 = 100$ generations.

Ex. 13.
```
$ sfs_code 3 1 -Td 0.0 P 0 .1 -Tg 0 P 0 2 \
    -TD 2.5 0 1 0.1 100 -TD 2.5 0 2 0.01 10 \
    -Tg 2.5 P 1 10 -Tg 2.5 P 2 15 -TE 2.6
```

Let's walk through this example step by step. First, `sfs_code 3 1` indicates that we are going to simulate a total of 3 populations for 1 iteration. Next `-Td 0.0 P 0 .1` indicates that there is going to be a demographic event at the end of the burn-in period for population 0. This demographic event will shrink the population to 1/10th its size. After the major contraction, `-Tg 0 P 0 2` indicates that population 0 will start exponentially growing at a rate of 2 per generation (the backslash '\' indicates that the command stretches onto the next line and can be ignored). Then, after 2.5 units of time, two new breeds are formed from this ancestral breed. Population 1 is created by `-TD 2.5 0 1 0.1 100`, indicating that an allele at frequency 0.1 in the parental population was used to form a population of 100 individuals. Population 2 is created by `-TD 2.5 0 2 0.01 10`, indicating that an allele at frequency 0.01 is used to form a population of size 10. Both breeds then start growing at an exponential rate (population 1 at a rate of

10, while population 2 grows at a rate of 15). Then, after another 0.1 units of time (100 generations, or approximately 200-300 years), the simulation ends and we draw the default of 6 individuals from each population. This simulation takes less than 5 seconds on a 2.33 GHz Intel Core 2 Duo MacBook Pro.

Migration

Individuals are free to migrate to any extant populations. The migration rate matrix indicates the average number of individuals in each population that are composed of individuals from each of the other populations. For the migration matrix $\mathbb{M}$, the (i, j) entry $m_{i,j}$ represents the expected number of individuals in population i that came from population j (this is also referred to as the "backward migration rate matrix"). To set the migration rate, you would use the command `--migMat` (`-m`). There are three ways to set the values of the migration matrix, indicated by the first argument to the option being either '`A`', '`P`', or '`L`'. You can set **All entries** to be the same value `M`:

```
--migMat (-m) A <M>
```

Note that this option specifies a symmetric island model, where the number of migrants into population i is M. So, for `<NPOP>`=3, there would be $M/2$ migrants from both of the other two populations. You can also set the migration rates explicitly **from one Population to another**:

```
--migMat (-m) P <Pto> <Pfrom> <M>
```

which would specify that the average number of migrants into population `Pto` from `Pfrom` is `M`. Finally, you can **List the entire migration matrix**:

```
--migMat (-m) L <M0,1>...<MNPOP,NPOP-1>
```

which would set each entry of the matrix. Note that the *diagonal entries are not specified*. For example, if you have 3 populations and want to use option 'L', you should specify all 6 entries: $M_{0,1}, M_{0,2}, M_{1,0}, M_{1,2}, M_{2,0}, M_{2,1}$.

In SFS_CODE, a Poisson number of individuals are chosen to migrate from population j to population i each generation with expected value $M_{i,j}$. Each migrant out of population j will be male with probability pMaleMig. You can set the **male migration rate** using

--pMaleMig (-y) [P <pop>] <p_{male}>

By default, p_{male}=1-propFemale, corresponding to the proportion of males in the originating population. By default, this is 0.5, but you can **change the proportion of females in a population** using

--propFemale (-f) [P <pop>] <pf>

This can be set for all populations simultaneously, or for a given population explicitly.

B.4.4 Mutation Models

There are 6 mutation models built into SFS_CODE. The basic initiation of a mutation model is as follows.

--substMod (-M) <mod> [args]

Table B.2 outlines the models and arguments for this option. The most basic mutation model (<mod> = 0) was proposed by Jukes and Cantor (1969), and referred to as **JC69**. This model assumes that the rate of mutation is equal among all nucleotides. A simple modification of this model was proposed by Kimura

Table B.2: Mutation models: arguments for option `--substMod`

<mod>	[args]	description
0	$\emptyset$	**JC69** model of equal mutation rates to and from all nucleotides.
1	<ψ>	**JC69+CpG** Simple model of hypermutable CpGs, where <ψ> is the non-CpG rejection rate.
2	<κ>	**Kimura 2-parameter** model, with <KAPPA> the transition-transversion bias.
3	<κ> <ψ>	**K2P+CpG** combining model 1 and 2.
4	$\emptyset$	**ZG2003** the generalized K2P model, where each nucleotide has its own transition/transversion bias (all parameters inferred by Zhang and Gerstein (2003)).
5	$\emptyset$	**Context-Dependent** model, where the mutation rate at each nucleotide depends on both of its adjacent neighbors (all parameters inferred by Hwang and Green (2004)). This is the model `SFS_CODE` was named after.

(1980) to account for the observation that most mutations tend to be transitions (A↔G or C↔T). This model (<mod> = 2) adds another parameter (the transition/transversion bias, κ), and is referred to as the Kimura 2-parameter model (or just **K2P**). An extension of the K2P model would be to allow a transition/transversion bias for each nucleotide (*i.e.*, the rate of A→G is not equal to the rate of C→T). Zhang and Gerstein (2003) fit the parameters of such a model to human data. This model has been implemented in `SFS_CODE` as <mod> = 4.

One feature of mammalian genomes is the presence of hypermutable CpGs (due to the deamination of methylated C's that are immediately 5' of a G). `SFS_CODE` implements a CpG extension to both the JC69 model and the K2P model (<mod> = 1 and 3, respectively). This is implemented by rejecting mutations at non-CpG

sites with probability <PSI>. Given a non-CpG site is rejected, a new site will be picked to mutate until either finding a CpG or accepting a non-CpG site. Once accepting a site to mutate, it will either mutate to a new nucleotide randomly (in the case of <mod>=1) or to a transitional nucleotide at a rate equal to <κ> (in the case of <mod>=3). For substitution models 1, 2, and 3, the mutation parameters (ψ and κ) can also be set for a single population using the following option.

--KAPPA (-K) [P <pop>] <κ>

--PSI (-C) [P <pop>] <ψ>

The most detailed model that is implemented in SFS_CODE is <mod>=5. This is a full context-dependent substitution model, where the site-specific rate of mutation depends on both of its adjacent nucleotides. This accounts for mutation rate variation due to CpGs as well as other context-effects found by Hwang and Green (2004). Conditional on picking a site to mutate, the replacement nucleotide will also depend on the flanking nucleotides. Choosing a new site to mutate is done using an inverse-CDF method, where relative hit-probabilities are defined by the cumulative site-specific mutation rates.

More generally, any trinucleotide substitution model can be used by updating the 64×4 rate matrix Q in the file gencontextrate.h and recompiling the program.

While SFS_CODE is based on simulating finitely many sites, it is also possible to simulate data under a pseudo-infinitely many sites model. It is pseudo because multiple hits can occur, but no more than one mutation will be segregating at a site at any given time. This is specified using the following option.

--INF_SITES (-I)

Mutation Rate Variation Across Sites and Loci

Context-dependent mutation models impose mutation rate variation along a sequence. However, not all species show evidence for such a mutation process (*e.g.*, Drosophila), despite having mutation rate variation. For such species, mutation rate variation has in the past been modeled as a discretized Γ distribution across sites. `SFS_CODE` allows you to simulate under such a model, allowing both sites as well as loci to have a mutation rate scaled by a discretized Γ distribution (with mean 1). These are implemented in the following options.

`--rateClassSites (-V) [P <pop>] <n`$_{\texttt{classes}}$`> <`α`>`

`--rateClassLoci (-v) [P <pop>] <n`$_{\texttt{classes}}$`> <`α`>`

These options allow you to specify a certain number of mutation rate classes ($\texttt{n}_{\texttt{classes}}$), which will be drawn from a $\Gamma(\alpha, \alpha)$ distribution (having mean 1 and variance $1/\alpha$).

B.4.5 Selfing and Generation-Effects

`SFS_CODE` generally assumes that all populations are randomly mating (subject to their relative fitnesses). However, in plant species in particular, mating is not random, such that an individual may be more likely to self-fertilize than to mate with another (an ultimate form of inbreeding). To accommodate this, `SFS_CODE` allows the user to specify a selfing rate, `s`, for each population using the following option.

`--self (-i) [P <pop>] <s>`

Moreover, when simulating multiple species, it will not always be the case that they will have the same generation time. For example, today, humans have a longer

generation time than most other primates (especially the non-apes). To account for this, SFS_CODE provides a generation effect option.

--GenEffect (-G) <pop> <G>

For the generation effect, G must be an integer (≥ 1 or ≤ -2). If it is positive, then the indicated population will experience G rounds of mating each generation. If G is negative, then the indicated population will only have a round of mating every |G| generations. For example, setting G=2 would shrink the generation time by half (leading to 2 rounds of random mating every generation), while setting G=-2 would double the generation time (leading to one round of random mating every second generation). At least one population must have G=1.

B.4.6 Changing Parameters Over Time

In SFS_CODE, many of the parameters can be changed at any time during the course of the simulation. Table B.5 outlines all the options that have been implemented in SFS_CODE, and any option that has an asterisk in the short name (third column) can be changed (or initiated) at any time using the following option.

-T<short_name> <τ> [args]

This means that if you are, for example, studying domesticated rice, and want to model the transformation to a primarily selfing organism, you might consider a population which starts with a low selfing rate, but $2N$ generations ago became 99% selfing. This could be achieved as follows.

Ex. 14. `./sfs_code 1 1 -TE 1 -i 0.2 -Ti 0 0.99`

Example 14 would simulate data assuming the selfing rate was 0.2 until the burn-in time ended, at which point the selfing rate would change to 99%. The simulation would then end after another $2N$ generations.

When using `-T*`, the option retains all the functionality as described above and in Table B.5. For example, consider simulating 2 populations, that diverged $10\times2N$ generations ago (*i.e.*, human-chimp divergence). Suppose you wanted to model recent positive selection (*e.g.*, within the last $2N$ generations) in the human genome while ancestral populations and chimpanzee are completely neutral. You might try the following example.

Ex. 15. `./sfs_code 2 1 -TS 0 0 1 -TE 10 -TW 9 P 0 1 5 1 0`

Example 15 would generate 2 populations, which split at time $\tau = 0$, and evolve independently for $10\times2N$ generations (`-TE 10`). However after having diverged for $9\times2N$ generations, all new nonsynonymous mutations in population 0 would be advantageous with $\gamma = 5$. In this case, the command "`-TW 9 P 0 1 5 1 0`" literally means change the distribution of selective effects at time $\tau = 9$ for population 0 to `type`=1, γ=5, `p_pos`=1, and `p_neg`=0.

Keep in mind that timed events only work with the short names, so you could not use `-TselDistType`, for example.

B.5 The non-Effect of the Effective Population Size

One challenge you will face when running forward simulations, is to pick an effective population size. In coalescent theory, this is a non-issue, as the results have been derived for limiting cases when the population size tends to infinity while parameters tend to zero, such that the product stays constant (isn't it nice that all parameters in population genetics being scaled by the population size?). However, in forward simulations, the actual population size used can become an issue, depending on what you are trying to model.

This section shows you that in most situations, the actual population size doesn't matter. In general, you should do a few simulations with larger and smaller population sizes to show that the population size used does not affect your simulations. Failing to do so could lead to a false interpretation of the results. However, it is always helpful to use the smallest population size possible, as this will make the simulations run quicker.

We will consider **varying the effective population size**. This is done using the following option.

`--popSize (-N) [P <pop>] <size>`

We will consider populations of size $N \in \{250, 500, 1000, 5000, 10000\}$. This should give us an idea of what is going on. We will consider just a couple of statistics: the distribution of the total number of polymorphisms and the average SFS. We will evaluate populations of constant size, as well as populations that have recently either grown or shrunk 10-fold (magnitude of change $\nu = 10$ or 0.1, approximately $0.1 \times 2N$ generations ago). We will also consider neutral models as well as selection under both `selDistType` 1 and 2 (`-W 1 5 0.1 0.8` and `-W 2 0.1 50 10 0.23 0.1279`, respectively). Considering just the case of no recombination yields 45 combinations (5 population sizes $\times$ 3 demographic models $\times$ 3 selective effects). The general command line looks like the following example (note that we are sampling 20 diploid individuals, or 40 chromosomes).

Ex. 16. `$ ./sfs_code 1 2000 -n 20 -N <`N`> -Td 0 <`ν`> -TE <`τ`> \`
`-W <type> [args]`

Figure B.4 summarizes the results. As you can see, for these demographic and selective effects, **the effective population size used has no impact on the distribution of SNPs or the SFS**. For the cases shown here, the only reason the curves do not perfectly overlap is because 2000 simulations is insufficient to

capture the true distribution. A very similar figure was generated (but not shown) with recombination ($\rho = 0.01$/site) with identical results.

B.6 Sampling From an Extinct Lineage

Recent advances in the extraction of ancient DNA has lead to the partial sequencing of the Neanderthal genome. This allows us to gain further insight into homo relatives, and actually learn more about our own species. One of the many questions that gains a lot of interest is whether or not humans mated with Neanderthals, and whether there is any evidence for or against it in our genomes. Understanding whether or not current statistical methods will have the power to detect evidence of such mating (or how much migration there would have had to have been to detect it) lies crucially in the hands of population genetic simulations.

Assuming that you've been able to simulate more than one population, sampling from an extinct lineage is actually dead easy (pun completely intended). You must simply "kill" one of the populations using the option `-TE <tau> [pop]`. Take the following example.

Ex. 17. `$ ./sfs_code 2 1 -TS 0 0 1 -TE 0.1 0 -TE 0.5`

During the burn-in, a single population would reach stationarity. At the end of the burn-in, the populations would allopatrically split (`-TS 0 0 1`). After $0.1 \times 2N$ generations, population 0 would effectively die (simulations for this population would stop). All the individuals from this population are still in memory, but evolution in this population ceases (all individuals would, quite literally, remain frozen in their prehistoric state). After an additional $0.4 \times 2N$ generations, the simulation completes. At completion, the default of 6 individuals will be sampled

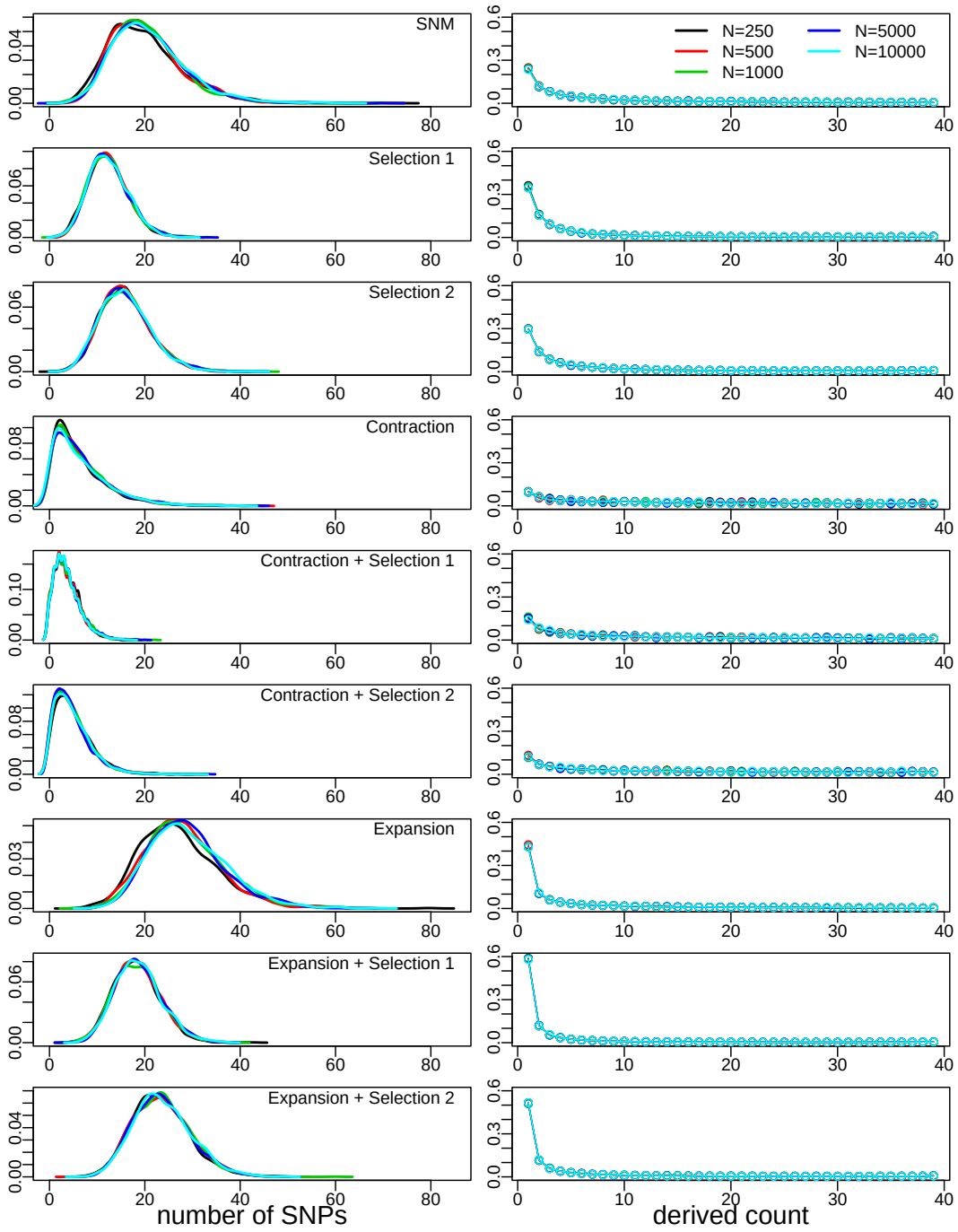

Figure B.4: The non-effect of the effective population size in `SFS_CODE`. Each panel has 5 curves, corresponding to different values of N_e (shown in legend in the upper-right plot). Each row corresponds to a different set of assumptions (demography/selection). Left is the distribution of the number of segregating sites, and the right is the average SFS across 2000 simulations.

from the extinct lineage (population 0) and 6 individuals will be sampled from the other population (you could consider this the human population).

For a slightly more detailed simulation, consider the human-Neanderthal-chimpanzee alignment, where only a single individual is chosen from each species. This could be implemented as follows.

Ex. 18. `$ ./sfs_code 3 1 -TS 0 0 1 -TS 8 1 2 -TE 9 2 -TE 10 -n 1`

Step-by-step, this would perform a single simulation of 3 populations (`./sfs_code 3 1`). At the end of the burn-in, the ancestral population splits into two populations (*e.g.*, the chimp and human-Neanderthal ancestor: `-TS 0 0 1`). After $8 \times 2N$ generations, humans and Neanderthals split (`-TS 8 1 2`). After another $2N$ generations, the Neanderthal population suddenly goes extinct (`-TE 9 2`). Finally, after a total of $10 \times 2N$ generations, the simulation ends, so we sample 1 individual from each species (including the extinct one).

Rather than have the Neanderthal population suddenly go extinct, it might be more useful to have them die out at an exponential rate, or something. Just as in section B.4.1, we could add a growth (or death) rate of population 2 to -2 at time $\tau = 8.5$ using `-Tg 8.5 P 2 -2`.

After simulating data, one could use `convertSFS_CODE` to analyze patterns of shared haplotypes, patterns of ancestral and derived alleles, etc. One could also consider humans and Neanderthals to be two populations with very high rates of migration, but negative selection acting strongly on the Neanderthal population until they are "bred out".

B.7 Using SFS_CODE on a Cluster

B.7.1 Your own Cluster

Any large scale simulation study cannot be performed without a large number of processors. Even coalescent simulations require a cluster in some situations, such as when using approximate likelihood techniques [*e.g.*, Hernandez et al. (2007a); Caicedo et al. (2007)]. However, if you do not have access to a cluster, or are unfamiliar with how to run jobs on a cluster, then this section will be insufficient for you (try the next section, which deals with using the CBSU cluster at Cornell University).

Being a forward simulation, SFS_CODE works well when identical jobs are sent to multiple processors, as they are all run independently. However, because the burn-in period takes a considerable amount of time, it is often beneficial to run several iterations on each processor (to take advantage of the short successive draws after the burn-in, as shown in Figure B.1). It therefore becomes a balance between the number of iterations to run on each processor versus the number of processors at your disposal to provide the most efficient results.

Let's consider the case where you want to run 2000 iterations of a single simulation. One way to complete the task is to split it up into 200 jobs, each of which runs 10 iterations. Each of the 200 jobs can then be submitted to a different processor successively. When all the processors have written their output, you could concatenate all 200 output files into a single file for analysis.

Setting the Seed

When running simulations on several processors simultaneously, there is a very nontrivial probability that some jobs will start at the same time. These jobs would

then have the same seed for the random number generator, and would therefore produce *exactly* the same results. This is very bad, every simulation must have a different seed. For any given simulation, you can **change the seed** using the following option.

```
--seed (-s) <value>
```

Depending on the configuration of your cluster, you may be able to generate a `SEEDFILE` with 200 lines (using `perl` or the system defined `RANDOM` shell variable), each line containing a unique seed that each process can pluck from. Alternatively, you may need to generate a `TASKFILE` with 200 lines, each line containing the entire `SFS_CODE` command, but with a unique seed. In the latter situation, you could use another program (written using `MPI`) to dynamically distribute each of the tasks across the available processors (not provided, but these so called "master-slave" algorithms are commonly used and can be downloaded from other sources).

Setting the seed is also useful if you ever want to reproduce a set of simulation results exactly.

Distributing the Work

Included in the distribution of `SFS_CODE` is an example `perl` script (`genSEEDS.pl`) for generating unique seeds (the `SEEDFILE`), an example `perl` script (`makeTask.pl`) for creating a `TASKFILE`, and an example `shell` file (`run_sfs_code_array.sh`) that could be used to run an *array* of jobs on a Sun Grid Engine Cluster. See comments in these files for more details. If you use something other than the Sun Grid Engine (SGE), then you are on your own (unless you are using the CBSU clusters described in the next section), as I am probably not familiar with it.

Assuming that you have already compiled the program and know where it is located on the cluster, you should be ready to generate some data. Change

directory to the folder containing the 3 files mentioned in the previous paragraph. The basic procedure would be to generate the SEEDFILE:

```
perl genSEEDS.pl <Nseeds>
```

where <Nseeds> is the number of simulations you want to perform. Next, if you want to do several simulations, with each one varying a certain number of parameters, it might be helpful to make a TASKFILE:

```
perl makeTask.pl
```

Note that makeTask.pl will have to be updated to incorporate the parameters that you are interested in. Next, you will want to update the shell file run_sfs_code_array.sh to either extract information from the TASKFILE or to contain the specific SFS_CODE command that you want to simulate. Finally, you will want to submit the shell file to the cluster.

B.7.2 Using SFS_CODE on the CBSU Cluster

SFS_CODE has a dedicated webpage at http://cbsuapps.tc.cornell.edu/sfscode.aspx. From here, it is possible to submit your simulation jobs to the cluster hosted by the Cornell University Computation Biology Service Unit (CBSU). The interface is simple. Enter your email address (to have a link with simulation results emailed to you), and a single set of SFS_CODE arguments. All you have to do is indicate the total number of populations you want to simulate (Npop), the total number of successive simulations that you would like to run on each process (Niter), the total number of repetitions you want to run (Nrep), and the total number of nodes you want to use. Then, copy-paste your command-line arguments into the text box. A total of Nrep×Niter simulations will be run automatically,

each with a different seed. You can only paste a single line into the text box. Running several different sets of simulations requires submitting several jobs.

The command will then be distributed across the requested number of processors. Upon completion, output will be concatenated into a single file, zipped, and stored on a server until downloaded (a link to the location will be emailed to you). Depending on the number of jobs in the queue, your job may take up to a few days to begin.

The CBSU clusters have a 24 hour time limit. This means that any job that you submit to the cluster must be completed within 24 hours. It is good practice to become familiar with the length of time that your job will take by running a few iterations locally (also helpful to ensure that your command-line works). You could then multiply the average amount of time per job by the total number of tasks you wish to perform and divide by the total number of processors. If this is greater than 24 hours, consider splitting the job into two or more sets.

B.8 Understanding the Output

A lot of information is stored during the course of an `SFS_CODE` simulation that will be useful in many different situations (yet useless in others). Unfortunately, this makes for a lot of output. In order to make the output file as concise as possible, a fairly complicated format is needed. However, I've also produced `convertSFS_CODE`, a program that will convert the `SFS_CODE` output to a more useful format (see next section). However, it is important to know all of the information that is contained in the output in the event you want to perform a type of analysis that has not yet been implemented in `convertSFS_CODE`. An example of the output looks something like the following:

```
./sfs_code 1 2 -L 2 66
SEED = -382034166
//iteration:1/2
>locus_0
GTTCCAGGAAGCTGGACAGTCTCTTATGGCGACATGGTAAATAAATTTGCGGTCCTGAAATCGGGT
>locus_1
AATGGGTTAGATTATGATTATGTGCATCGTCTTTACACGAGTGGAGTCATTCGACTTTTCGTACTA
Nc:500;
MALES:3;
0,A,24,-237,0,TTA,A,1,Y,N,0.0,2,0.1,0.8;
//iteration:2/2
>locus_0
TTTTCAGTCTGTTTTGTCAAAGATTATTCTTTTGGGCTCCTCACGCACCTTAAGAAGTGTATATAC
>locus_1
TACGCTATCAACTACAATATACATAGTGTGGTTTTCGATGGCCTTAGGTCAGTTGACCTACGTAAC
Nc:500;
MALES:3;
1,A,28,-523,0,GTG,A,1,V,E,0.0,2,0.1,0.3;
```

The first line has the command line used to call `SFS_CODE`. In this case I've asked `SFS_CODE` to generate two iterations of a single population, with two loci, each of length 66 basepairs. The second line includes the value of the seed for the random number generator (this can be used to reproduce the results, though the version used to generate this output may be different from the one you are using so you may not be able to reproduce this output exactly). The third line starts the results of the simulations. Each iteration that is simulated starts with "`//iteration:`", followed by the iteration number and the total number of iterations being performed. The fourth line starts a fasta-style representation of the nucleotide sequences at each locus. The next line reports the final population size (N_c) of each population in a comma delimited list terminated with a semi-colon. The next line ("`MALES:`") provides the *index* for the first male that was sampled (*not the number of males in the sample*), in this case indicating that individuals 3, 4, and 5 are male while individuals 0, 1, and 2 are female (note that these are diploid individuals, so chromosomes 0-5 belong to females and chromosomes 6-11

belong to males). The next line starts the information regarding every mutation that contributes information to the sampled sequences in a comma delimited list terminated with a semi-colon. There are at most 20 mutations per line, so mutations can span several lines. The information provided for each mutation are as follows:

1. locus that the mutation arose on (zero-based)
2. 'A', 'X', 'Y', indicating Autosomal, X-, or Y-linked mutation, respectively
3. position of mutation in locus (zero based)
4. generation mutation arose (negative for mutations arising during burn-in)
5. generation mutation fixed in population (or time of sampling if segregating)
6. ancestral trinucleotide (middle nucleotide mutated, NOT CODON)
7. derived nucleotide
8. 0 or 1 for synonymous or nonsynonymous (respectively; 0 for non-coding)
9. ancestral amino acid (single character representation; '`X`' for non-coding)
10. derived amino acid (single character; '`X`' for non-coding)
11. fitness effect (this is s, NOT $\gamma = PNs$)
12. number of chromosomes (n) that carry the mutation

13+ comma delimited list of the n chromosomes carrying mutation

Each mutation event is terminated with a semi-colon. The list of chromosomes carrying each mutation is reported as a decimal: $p.c$, where p is the population number (zero based) and c represents the chromosome number in that population (also zero based). Take the mutation event reported in the first iteration: "`0,A,24,-237,0,TTA,A,1,Y,N,0.0,2,0.1,0.8;`". This indicates that it occurred at the first locus (zero), which is autosomal, at position 24. This mutation arose

237 generations before sampling (*i.e.*, 237 generations before the burn-in period ended), and was segregating at time of sampling (sampled at time 0). This mutation was a nonsynonymous mutation having no fitness effect, and was carried by two chromosomes (1 and 8) in population zero (the only population simulated).

Mutations that fix in the population during the burn-in period are not recorded. If you want to track fixations while simulating a single population, use the `-TE` option. Mutations that are fixed in the sample from population p will be reported as p.-1. It is possible to distinguish mutations that are fixed in the sample but segregating in the population using the 4th entry (the generation it supposedly fixed). If it "fixed" at the end of the simulation, then the fixation time will be the same as segregating polymorphisms. This can only happen if it was still segregating in the population (as random mating would not yet have occurred).

It is generally encouraged to retain the ancestral sequence (just in case you want to go back and re-analyze some previous simulations and need the actual sequences; *e.g.*, to setup the observed McDonald-Kreitman tables). However, they can take up a lot of space in the output, so you can exclude ancestral sequences using the following option.

`--noSeq (-A)`

By default, output will be printed to the screen. If you would rather it be written to a file, you can use the following option.

`--outfile (-o) [a] <file>`

The optional character '`a`' would allow you to append to a file rather than overwriting it. When multiple iterations are run, and output is being directed to a file, the progress of the simulation will be printed to the error file. By default, the error file is the screen, but this can be changed using the following option

```
--errfile (-e) [a] <file>
```

This is also useful for keeping track of any error messages that arise (such as when figuring out what might have gone wrong with a set of command line arguments).

B.9 Using convertSFS_CODE to Generate Useful Data

The output produced by the program SFS_CODE is fairly concise, but is not the easiest file to parse. I have therefore provided the additional program convertSFS_CODE, which takes the output from SFS_CODE, and converts it to a format that might be easier for you to use. These include various summary statistics, as well as the format required for the program STRUCTURE [*e.g.*, Falush et al. (2003)], and an output analogous to format used by the coalescent simulation program ms (Hudson, 2002), among others.The basic usage of convertSFS_CODE is as follows:

```
./convertSFS_CODE <input_file> <option [args]>
```

The <input_file> to convertSFS_CODE is the output file from SFS_CODE. The options available are outlined in Table B.3. As an example, consider generating a human-neandertal-chimpanzee alignment using a single chromosome from each. You might consider the following slightly modified version of Example 18.

Ex. 19.
```
./sfs_code 3 1 -L 1 66 -TS 0 0 1 -TS 8 1 2 -TE 9 2 \
    -TE 10 -n 1 -o out.txt
```

This would generate a single simulation of 3 individuals for a locus of length 66 basepairs. You could then use convertSFS_CODE to generate a **fasta-style alignment** of one chromosome using the following example.

Ex. 20.
```
./convertSFS_CODE out.txt -a I 1 0
```

This would print the alignment of three chromosomes to the screen. It might look like the following:

```
>it0pop0ind0locus0
GTATGTATAGGGCTTGGTATTGAAAATAGGTCCCAGGAAATCTTGACCGGCACCCAAGAGGTCATG
>it0pop1ind0locus0
GTATGTATAGGGCTTTATATTGAAAATAGGTCCCAGGAAATCTTGACCGGCACCCAAGAGGTCATG
>it0pop2ind0locus0
GTATGTATAGGGCTTTATATTGAAAATAGGTCCCAGGAAATCTTGACCGGCACCAAAGAGGTCATG
```

The name of each sequence indicates the iteration ("`it0`"), the population ("`pop0`"), the individual chromosome ("`ind0`"), and the locus ("`locus0`"), followed by the actual sequence on the next line. If you just wanted to extract a single sequence from human and chimpanzee, then you would use the "`P.I`" option, which would allow you to select certain chromosomes from certain populations. In this case, you would use "`P.I 2 0.0 1.0`" to indicate that you want two chromosomes to be printed: chromosome 0 from population 0 and chromosome 0 from population 1.

Ex. 21. `./convertSFS_CODE out.txt -a P.I 2 0.0 1.0`

If you also want to include the true ancestral sequence of each population to be printed, include the character '`A`'. Note that this is the ancestral sequence of a population, and not the ancestral sequence of a pair or group of populations. It will be printed just as any other sequence from a population, but will have "`indA`", such that the chromosome identifier is '`A`'.

To generate the **McDonald-Kreitman table** for the human-chimp simulation, you would use the following example.

Ex. 22. `./convertSFS_CODE out.txt --MK 1 0`

This indicates that you want to use population 1 for polymorphism (the human population) and population 0 as an outgroup to call fixed differences (chimp polymorphism is not included). The output for this particular case is as follows: `0 0 0 2 1.0000`. Not very interesting. There were (in order of appearance) zero synonymous and nonsynonymous polymorphisms, zero synonymous fixed differences, and two nonsynonymous fixed differences. The last value given is the Fisher exact test p-value for the table. You can print the table to a file using the flag "`F <filename>`". If you ran several iterations, you can print each one independently using "`ITS -1`". More generally, if you just want to print the first n iterations, replace "-1" with "n", the -1 just allows you to not worry about the actual number of simulations that were done. By default, the outgroup sample size is 1 (*e.g.*, using just the reference sequence to call fixed differences), but this can be changed to n using the option "`OGSS` n". This is helpful if you want to test the effect of using multiple outgroup sequences on calling fixed differences (many sites that are apparently fixed between populations are actually the result of sampling a common mutation segregating in one of the populations). Finally, if you want to use parsimony to call synonymous and nonsynonymous mutations (instead of using their true classification stored during the mutation), use "`OBS`". As an example, suppose that the file `out.txt` contained several simulations from two populations across 3 loci, but we wanted the observed (*i.e.*, using parsimony) MK table for the first locus using a sample size of 2 for the outgroup. We could use the following example:

Ex. 23.
```
./convertSFS_CODE out.txt --MK 1 0 F mk.txt ITS -1 \
    L 1 0 OGSS 2 OBS
```

For printing the **site-frequency spectrum** (**SFS**), use `--SFS`. There are many similar options for this output. However, you can also specify the SFS of just

autosomal (`A`), X or Y linked SNPs (`X` or `Y`, respectively). You can also specify the type of mutations to include in the SFS (*e.g.*, synonymous, nonsynonymous, or both). In this case, you would use "`T 0`", "`T 1`", or "`T 2`" (respectively). Non-coding mutations are considered synonymous, and will not be reported for "`T 2`". Sometimes using an outgroup to identify the ancestral state of a polymorphism under parsimony will be wrong (Hernandez et al., 2007b). By default, the true SFS will be generated, but you can also use parsimony (if you've simulated at least two populations) by including "`OBS <ing> <og> [og`$_{\texttt{size}}$`]`", where `ing` is the in-group used for polymorphism, `og` is the outgroup used for identifying the ancestral states of polymorphisms, and the optional argument `og`$_{\texttt{size}}$ indicates the number of sequences to use from the outgroup (default is 1).

To produce output in a similar format as the coalescent simulator **ms** (Hudson, 2002), you can use the option `--ms`. You can print the output to a file using `F <file>`, and specify the type of mutations to output (either synonymous/non-coding, `T 0`; nonsynonymous, `T 1`; or both `T 2`).

Finally, you can produce the input file for the Bayesian clustering algorithm **STRUCTURE** (Falush et al., 2003) using the option `--structure`. For this option, you can specify the centimorgans per megabase (using `CMMB <c>`), and restrict the output to either specific individuals (`I <n> ...`), specific populations (`P <n> ...`) or specific individuals from specific populations (`P.I. <n> ...`).

It is important to note that you can use several options at once. For example, if you want to generate both MK tables and the SFS, you can put them both in the command line. Additionally, if you simulated several species, and want to generate the observed SFS using species with increasing divergence, then you can just concatenate all your `--SFS` commands as follows:

Ex. 24. `./convertSFS_CODE out.txt --SFS OBS 0 1 F sfs1.txt \`

`--SFS OBS 0 2 F sfs2.txt --SFS OBS 0 3 F sfs3.txt`

Table B.3: `convertSFS_CODE` Options. Any combination of `<arguments>` can be used (in any order) for a given task.

long	short	<arguments>	description
--help	-h	$\emptyset$	**Display help menu**
--alignment	-a	**Print sequence alignment in fasta format**	
		[A]	Print ancestral population sequence
		[F <file>]	Print sequences to a file
		[L <n> <L_1>...<L_n>]	Only print n loci.
		[I <n> <I_1>...<I_n>]	Only print n chromosomes (but from all populations).
		[P <n> <P_1>...<P_n>]	Only print specific populations.
		[P.I <n> <$p_1.c_1$>...]	Only print specific chromosomes from specific populations
		[ITS <n>]	Only print first n iterations
--ms	-m	**Produce ms-style output**	
		[F <file>]	Print MK tables to a file
		[T <type>]	Extract mutations of <type> = 0, 1, or 2 (synon., nonsynon., or both)
--MK	-M	**Print McDonald-Kreitman tables**	
		<ing> <og>	Ingroup and outgroup (**required**)
		[F <file>]	Print MK tables to a file
		[ITS [n]]	Print each iteration [or just first n]
		[L <n> <L_1>...<L_n>]	Only print n loci.
		[OGSS <size>]	Set the outgroup sample size
		[N <n>]	Randomly sample n chromosomes from each population
		[OBS]	Use parsimony to call synonymous and nonsynonymous mutations

Continued on next page...

Table B.3 (Continued)

long	short	<arguments>	description
--structure	-s	**Print structure input**	
		[F <file>]	Print to a file
		[CMMB <c>]	set centiMorgans/Megabase to c
		[I <n> <I_1>...<I_n>]	Only print n chromosomes (but from all populations).
		[P <n> <P_1>...<P_n>]	Only output specific populations.
		[P.I <n> <$p_1.c_1$>...]	Only print specific chromosomes from specific populations
--SFS	-S	**Print site-frequency spectra (SFS)**	
		[A]	Extract only autosomal loci
		[F <file>]	Print SFS to a file
		[ITS [n]]	Print each iteration [or just first n]
		[L <n> <L_1>...<L_n>]	Only print n loci.
		[N <n>]	Randomly sample n chromosomes from each population
		[OBS <ing> <og> [og_{size}]]	Use parsimony to identify ancestral alleles from an outgroup [using og_{size} chromosomes]
		[I <n> <I_1>...<I_n>]	Only print n chromosomes (but from all populations).
		[P <n> <P_1>...<P_n>]	Only output specific populations.
		[T <type>]	Extract mutations of <type> = 0, 1, or 2 (synon., nonsynon., or both)
		[X]	Extract only X-linked mutations
		[Y]	Extract only Y-linked mutations

B.10 Default Parameter Values

Running `SFS_CODE` with the default parameters will generate a sample of six diploid individuals (12 chromosomes) under the standard neutral model assumptions of a constant population size, absence of selection, etc. Every parameter discussed below can be changed using the command line options described in Table B.5. The full list of default parameter values are given in Table B.4.

Table B.4: Default parameter values used in `SFS_CODE`.

Parameter	**value**
Effective population size	500
Ploidy	2
Allo- (0) or Auto-ploidy (1)	0
Relative size of female population	0.5
Sample size for each population	6
Migration between populations	0.0
Probability male migrates (if migration)	0.5
θ/site	0.001
Substitution model	0
Number of mutation rate classes (sites)	1
Number of mutation rate classes (loci)	1
Infinite sites?	0
ρ/site	0.0
Number of loci	1
Length of locus (bp)	5001
Linkage within loci	0.0
Annotation	`C`
Sex chromosome?	no
Self-fertilization rate	0.0
Selection distribution	0
Proportion of loci subject to selection	1.0
Non-lethal mutation rate	1.0
Initial burn-in period ($\times$PN)	5
Burn-in period of subsequent iterations	2
Print ancestral sequence?	yes

B.11 Summary of Options and Arguments

In SFS_CODE, an **option** is a feature that allows you to change the characteristics of the simulation. Every option implemented in SFS_CODE is summarized in Table B.5. There are basically five types of options: (1) those that control the output, (2) those those that affect all populations or set foundation for the simulation ("Global Options"), (3) those that may be population specific ("Population Options"), (4) those that describe the effect of natural selection, and (5) those that govern the demographic events ("Evolutionary Events"). All options (except evolutionary events) have both a **long name** (preceded by two dashes) and a **short name** (a single character preceded by a single dash), which can be used interchangeably (*e.g.*, you could either use "`--theta 0.01`" or "`-t 0.01`" to set the population scaled mutation rate to 0.01 for all populations). Most options have **arguments**. Arguments that are **required** are enclosed in <angled brackets>. Arguments that are **optional** are enclosed in [square brackets].

Some options apply to all populations (*e.g.*, the length of the simulated sequence `-L`), and some apply only to a specific population (such as the generation effect `-G`). Others default to all populations, but allow you to specify a specific population. These are denoted by `[P <pop>]` in the argument list. This means that you can add the character '`P`' followed by the number of a specific population to apply the option to that population exclusively. For example, if you are simulating two populations, one of which is twice the size of the other, you might add "`-N P 1 1000`" to your command line.

Generally speaking, options can be used in any order. The exceptions are `--linkage`, `--annotate`, and `--sex`, which must come after `-L` if `-L` is used (if `-L` is not used, then order really doesn't matter). However, arguments for each option are in a specified order, and must always be used in the proper order.

Table B.5: SFS_CODE Options

	long	short[a]	<arguments>	description
Output	--help	-h	∅	Display help menu. (p80)
	--noSeq	-A	∅	DO NOT print ancestral sequences. (p115)
	--outfile	-o	[a] <file>	Write [or <a>ppend] output to a <file> instead of the screen. (p115)
	--errfile	-e	[a] <file>	Write [or <a>ppend] error messages to a <file>. (p116)
Selection	--selDistType	-W*	[P/L] <type> [arg]	Distribution of selective effects. (p91, Table B.1)
	--neutPop	-w*	<pop>	No selection on population <pop>. (p93)
	--constraint	-c*	[P/L] <f_0>	Set the proportion of nonlethal mutations to <f_0>. (p95)
Global Options	--BURN	-B	<burn>	Burn-in time for initial population. (p82)
	--BURN2	-b	<burn>	Burn-in time for subsequent simulations. (p82)
	--length	-L	<R> <L_1> [<L_2>..] [R]	Simulate <R> loci (regions) with lengths <L_1> [<L_2>... opt.; add 'R' after a subset of lengths to repeat]. (p83)
	--linkage	-l	<p/g> <d_1> [<d_2>..] [R]	Set linkage between adjacent loci (either <p>hysical, or <g>enetic distance) to <d_1> [<d_2> ..<d_{R-1}> opt.; add 'R' to repeat pattern]. MUST USE AFTER -L. (p83)
	--annotate	-a	<a_1> [<a_2>..<a_R>] [R]	Annotate each locus as C (coding) or N (non-coding) [<a_2>..<a_R> opt; add 'R' to repeat pattern]. Note: If coding (default), length will be rounded to nearest codon. (p84)
	--sex	-x	<x_1> [<x_2>...] [R]	Annotate sex loci, either 0 or 1 (autosomal or sex, resp.). Use only <x_1>, or specify each locus, or put 'R' at end to repeat a subset. Only implemented for the diploid case (P=2). Males do not recombine at sex loci. (p84)

Continued on next page...

Table B.5 (Continued)

	long	short[a]	<arguments>	description
	--ploidy	-P	<ploidy>	Set the ploidy of all populations to <ploidy> (1,2,4 only; *i.e.*, haploid, diploid, or tetraploid). (p87)
	--tetraType	-p	<0/1>	If P=4 (tetraploid) assume auto- or allotetraploid (0 or 1, resp.). (p87)
	--substMod	-M	<mod> [args]	Set the mutation model. (p99, Table B.2.)
	--INF_SITES	-I	$\emptyset$	Avoid multiple mutations segregating at the same site concurrently. Multiple hits can still occur for long divergence times. (p101)
	--seed	-s	<int>	Set random number seed to <int>. This should always be set manually if using this program on a cluster!! (p110)
Population Options	--theta	-t*	[P <p>] <θ>	Set the PER SITE population scaled mutation rate to <θ> for ALL populations [or just population <p>]. (p81)
	--rho	-r*	[P <p>] <ρ>	Set the PER SITE population scaled recombination rate to <ρ> for ALL populations [or just population <p>]. (p81)
	--popSize	-N*	[P <p>] <size>	Set all population to size <size> 'P'-ploid individuals [or just population <p>]. (p85)
	--sampSize	-n	<SS_1>..<SS_{NPOP}>	Sample <SS_1> individuals from population 1, ..., <SS_{NPOP}> individuals from population NPOP. Use the value -1 to sample an entire population. (p89)
	--migMat	-m*	A <M>	Set the migration rate to/from all pops to <M>/(NPOP-1). (p98)
			P <P_{to}> <P_{from}> <M>	Set the migration rate entry $m_{\text{to,from}}$ = <M>
			L <$M_{0,1}$>...<$M_{NPOP,NPOP-1}$>	Set all entries of the migration matrix.

Continued on next page...

Table B.5 (Continued)

	long	short[a]	<arguments>	description
	--pMaleMig	-y*	[P <p>] <p_{male}>	Set the proportion of migrants out of each population [or just population <p>] that are male to <p_{male}>. (p99)
	--propFemale	-f	[P <p>] <pf>	Set the proportion of females in each population [or just <pop>] to <pf>, default 0.5. (p99)
	--self	-i*	[P <p>] <s>	Set the selfing rate <s> for ALL populations [or just population <p>]. (p102)
	--KAPPA	-K*	[P <p>] <κ>	Set transition/ transversion rate ratio <κ> (only valid for --substMod 2 or 3). (p101)
	--PSI	-C*	[P <p>] <ψ>	Set CpG bias parameter (non-CpG rejection rate; only valid for --substMod 1 or 3). (p101)
	--rateClassSites	-V	[P <p>] <$n_{classes}$> <α>	Mutation rate variation among sites in loci (discrete Gamma model of <$n_{classes}$> classes with rate <α> and mean 1). (p102)
	--rateClassLoci	-v	[P <p>] <$n_{classes}$> <α>	Mutation rate variation among loci (discrete Gamma model of <$n_{classes}$> classes with rate <α> and mean 1). (p102)
	--GenEffect	-G*	<pop> <G>	Generation time effect. <G> $>$ 1 makes <pop> experience <G> rounds of mating each generation while if <G> < -1 mating occurs only every |<G>| generations. <G> must be an integer, with <G> > 0 or <G> < -1. (p103)
Evolutionary Events		-Td	<τ> [P <p>] <ν>	Discrete population size change at time <τ> with magnitude <ν> = N_{new}/N_{old}, where N_{old} is the size of <pop> prior to the event, NOT ANCESTRAL! (p85)
		-Tg	<τ> [P <p>] <α>	Set the exponential growth rate of a population to <α> at time <τ>. (p85)

Continued on next page...

Table B.5 (Continued)

	long	short[a]	<arguments>	description
		-Tk	<τ> [P <p>] <K> <r>	Logistic growth at rate <r> beginning at time <τ> until final population size <K> is reached. (p86)
		-TE	<τ> [p]	Terminate simulation [or just a population] at time <τ> $\times PN0$. (p87)
		-TS	<τ> <i> <j>	Split population <i> at time <τ>$\times PN$ generations to found population <j>. (p96)
		-TD	<τ> <i> <j> <f> <N> [l]	Domesticate population <j> with <N> individuals from <i> at time <τ> using a derived allele at frequency <f>$\pm$5%. (p96)

[a]Asterisk in short name indicates that the parameters can be changed (or option initiated) at any time using -T<short_name> <τ> <args>. See Section B.4.6 on page 103.

BIBLIOGRAPHY

Abegg, C. and Thierry, B. (2002). Macaque evolution and dispersal in insular south-east asia. *Biol J Linn Soc*, 75:555–576.

Akashi, H. (1999). Inferring the fitness effects of DNA mutations from polymorphism and divergence data: statistical power to detect directional selection under stationarity and free recombination. *Genetics*, 151(1):221–238.

Akashi, H. and Schaeffer, S. W. (1997). Natural selection and the frequency distributions of "silent" DNA polymorphism in Drosophila. *Genetics*, 146(1):295–307.

Andolfatto, P. (2005). Adaptive evolution of non-coding DNA in Drosophila. *Nature*, 437(7062):1149–1152.

Arndt, P. F., Petrov, D. A., and Hwa, T. (2003). Distinct changes of genomic biases in nucleotide substitution at the time of Mammalian radiation. *Mol Biol Evol*, 20(11):1887–1896.

Balloux, F. and Goudet, J. (2002). Statistical properties of population differentiation estimators under stepwise mutation in a finite island model. *Mol Ecol*, 11(4):771–783.

Barrett, J. C., Fry, B., Maller, J., and Daly, M. J. (2005). Haploview: analysis and visualization of ld and haplotype maps. *Bioinformatics*, 21(2):263–265.

Baudry, E. and Depaulis, F. (2003). Effect of misoriented sites on neutrality tests with outgroup. *Genetics*, 165(3):1619–1622.

Belle, E. M. S., Duret, L., Galtier, N., and Eyre-Walker, A. (2004). The decline of isochores in mammals: an assessment of the GC content variation along the mammalian phylogeny. *J Mol Evol*, 58(6):653–660.

Bernardi, G. (2000). Isochores and the evolutionary genomics of vertebrates. *Gene*, 241(1):3–17.

Bernardi, G., Hughes, S., and Mouchiroud, D. (1997). The major compositional transitions in the vertebrate genome. *J Mol Evol*, 44 Suppl 1:S44–S51.

Blake, R. D., Hess, S. T., and Nicholson-Tuell, J. (1992). The influence of nearest neighbors on the rate and pattern of spontaneous point mutations. *J Mol Evol*, 34(3):189–200.

Boyko, A. R., Williamson, S. H., Indap, A. I., Degenhardt, J. D., Hernandez, R. D., Lohmueller, K. E., Adams, M. D., Schmidt, S., Sninsky, J. J., Sunyaev, S. R., White, T. J., Nielsen, R., Clark, A. G., and Bustamante, C. D. (2007). Quantifying the distribution of selective effects among newly arising amino acid mutations in the human genome. *Submitted*.

Bustamante, C. D., Fledel-Alon, A., Williamson, S., Nielsen, R., Hubisz, M. T., Glanowski, S., Tanenbaum, D. M., White, T. J., Sninsky, J. J., Hernandez, R. D., Civello, D., Adams, M. D., Cargill, M., and Clark, A. G. (2005). Natural selection on protein-coding genes in the human genome. *Nature*, 437(7062):1153–1157.

Bustamante, C. D., Wakeley, J., Sawyer, S., and Hartl, D. L. (2001). Directional selection and the site-frequency spectrum. *Genetics*, 159(4):1779–1788.

Caicedo, A. L., Williamson, S. H., Hernandez, R. D., Boyko, A., Fledel-Alon, A., York, T. L., Polato, N. R., Olsen, K. M., Nielsen, R., McCouch, S. R., Bustamante, C. D., and Purugganan, M. D. (2007). Genome-wide patterns of nucleotide polymorphism in domesticated rice. *PLoS Genetics*, 3(9):1745–1756.

Chimpanzee Sequencing and Analysis Consortium (2005). Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature*, 437(7055):69–87.

Clark, A. G., Hubisz, M. J., Bustamante, C. D., Williamson, S. H., and Nielsen, R. (2005). Ascertainment bias in studies of human genome-wide polymorphism. *Genome Res*, 15(11):1496–1502.

Cutler, D. J. (2000). Understanding the overdispersed molecular clock. *Genetics*, 154(3):1403–1417.

Delson, E. (1980). Fossil macaques, phyletic relationships and a scenario of deployment. In Lindburg, D., editor, *The macaques. Studies in ecology, behavior and evolution.*, chapter 2, pages 10–30. van Nostrand Rheinhold.

Dudek, S. M., Motsinger, A. A., Velez, D. R., Williams, S. M., and Ritchie, M. D. (2006). Data simulation software for whole-genome association and other studies in human genetics. *Pac Symp Biocomput*, pages 499–510.

Duret, L., Mouchiroud, D., and Gautier, C. (1995). Statistical analysis of vertebrate sequences reveals that long genes are scarce in gc-rich isochores. *J Mol Evol*, 40(3):308–317.

Duret, L., Semon, M., Piganeau, G., Mouchiroud, D., and Galtier, N. (2002). Vanishing GC-rich isochores in mammalian genomes. *Genetics*, 162(4):1837–1847.

Eberle, M. A., Rieder, M. J., Kruglyak, L., and Nickerson, D. A. (2006). Allele frequency matching between snps reveals an excess of linkage disequilibrium in genic regions of the human genome. *PLoS Genet*, 2(9):e142.

ENCODE Project Consortium (2004). The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640.

Ewens, W. J. (2004). *Mathematical Population Genetics*. Springer-Verlag, New York, 2 edition.

Eyre-Walker, A. (1993). Recombination and mammalian genome evolution. *Proc Biol Sci*, 252(1335):237–243.

Eyre-Walker, A. (1999). Evidence of selection on silent site base composition in mammals: potential implications for the evolution of isochores and junk DNA. *Genetics*, 152(2):675–683.

Falush, D., Stephens, M., and Pritchard, J. K. (2003). Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–1587.

Fay, J. C. and Wu, C. I. (2000). Hitchhiking under positive Darwinian selection. *Genetics*, 155(3):1405–1413.

Ferguson, B., Street, S. L., Wright, H., Pearson, C., Jia, Y., Thompson, S. L., Allibone, P., Dubay, C. J., Spindel, E., and Norgren, R. B. (2007). Single nucleotide polymorphisms (snps) distinguish indian-origin and chinese-origin rhesus macaques (macaca mulatta). *BMC Genomics*, 8:43.

Fooden, J. (1980). Classification and distribution of living macaques (macaca lcepede, 1799). In Lindburg, D., editor, *The macaques. Studies in ecology, behavior and evolution.*, chapter 1, pages 1–9. van Nostrand Rheinhold.

Fraser, A. S. (1957). Simulating of genetic systems by automatic digital computers. *Aust. J. Biol. Sci.*, 10:484–491.

Fryxell, K. J. and Zuckerkandl, E. (2000). Cytosine deamination plays a primary role in the evolution of mammalian isochores. *Mol Biol Evol*, 17(9):1371–1383.

Fu, Y. X. (1995). Statistical properties of segregating sites. *Theor Popul Biol*, 48(2):172–197.

Fu, Y. X. and Li, W. H. (1993). Statistical tests of neutrality of mutations. *Genetics*, 133(3):693–709.

Fullerton, S. M., Bernardo Carvalho, A., and Clark, A. G. (2001). Local rates of recombination are positively correlated with gc content in the human genome. *Mol Biol Evol*, 18(6):1139–1142.

Galtier, N. and Gouy, M. (1998). Inferring pattern and process: maximum-likelihood implementation of a nonhomogeneous model of DNA sequence evolution for phylogenetic analysis. *Mol Biol Evol*, 15(7):871–879.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman & Hall/CRC, Boca Raton, FL., 2nd edition.

Griffiths, R. C. and Tavaré, S. (1994). Sampling theory for neutral alleles in a varying environment. *Philos Trans R Soc Lond B Biol Sci*, 344(1310):403–410.

Griffiths, R. C. and Tavaré, S. (1998). The age of a mutation in a general coalescent tree. *Stochastic Models*, 14:273–295.

Griffiths, R. C. and Tavaré, S. (1999). The ages of mutations in gene trees. *Annals of Applied Probability*, 9:567–590.

Guillaume, F. and Rougemont, J. (2006). Nemo: an evolutionary and population genetics programming framework. *Bioinformatics*, 22(20):2556–2557.

HapMap (2005). A haplotype map of the human genome. *Nature*, 437(7063):1299–1320.

Hartl, D. L., Moriyama, E. N., and Sawyer, S. A. (1994). Selection intensity for codon bias. *Genetics*, 138(1):227–234.

Hayasaka, K., Fujii, K., and Horai, S. (1996). Molecular phylogeny of macaques: implications of nucleotide sequences from an 896-base pair region of mitochondrial dna. *Mol Biol Evol*, 13(7):1044–1053.

Hernandez, R. D., Hubisz, M. J., Wheeler, D. A., Smith, D. G., Ferguson, B., Rogers, J., Nazareth, L., Indap, A., Bourquin, T., McPherson, J., Muzny, D., R., Nielsen, R., and Bustamante, C. D. (2007a). Demographic histories and patterns of linkage disequilibrium in Chinese and Indian rhesus macaques. *Science*, 316(5822):240–243.

Hernandez, R. D., Williamson, S. H., and Bustamante, C. D. (2007b). Context dependence, ancestral misidentification, and spurious signatures of natural selection. *Mol Biol Evol*, 24(8):1792–1800.

Hernandez, R. D., Williamson, S. H., Zhu, L., and Bustamante, C. D. (2007c). Context-dependent mutation rates may cause spurious signatures of a fixation bias favoring higher GC-content in humans. *Mol Biol Evol*, 24(10):2196–2202.

Hess, S. T., Blake, J. D., and Blake, R. D. (1994). Wide variations in neighbor-dependent substitution rates. *J Mol Biol*, 236(4):1022–1033.

Hey, J. (2004). Fpg-a computer program for forward population genetic simulation. `http://lifesci.rutgers.edu/~heylab/HeylabSoftware.htm#FPG`.

Hey, J. (2005). On the number of new world founders: a population genetic portrait of the peopling of the americas. *PLoS Biol*, 3(6):e193.

Hoggart, C. J., Chadeau, M., Clark, T. G., Lampariello, R., Iorio, M. D., Whittaker, J. C., and Balding, D. J. (2007). Sequence-level population simulations over large genomic regions. *Genetics*.

Hudson, R. R. (1983a). Properties of a neutral allele model with intragenic recombination. *Theor Popul Biol*, 23(2):183–201.

Hudson, R. R. (1983b). Testing the constant-rate neutral allele model with protein sequence data. *Evolution*, 37:203–217.

Hudson, R. R. (1990). Gene genealogies and the coalescent process. *Oxf. Surv. Evol. Biol.*, 7:1–44.

Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338.

Hudson, R. R. and Kaplan, N. L. (1995). Deleterious background selection with recombination. *Genetics*, 141(4):1605–1617.

Hwang, D. G. and Green, P. (2004). Bayesian Markov chain Monte Carlo sequence analysis reveals varying neutral substitution patterns in mammalian evolution. *Proc Natl Acad Sci U S A*, 101(39):13994–14001.

Jabbari, K. and Bernardi, G. (1998). Cpg doublets, cpg islands and alu repeats in long human dna sequences from different isochore families. *Gene*, 224(1-2):123–127.

Jukes, T. H. and Cantor, C. R. (1969). Evolution of protein molecules. In Munro, H. N., editor, *Mammalian protein metabolism*, pages 21–132. Academic Press, New York.

Kent, W. J. (2002). BLAT–the BLAST-like alignment tool. *Genome Res*, 12(4):656–664.

Kim, Y. and Stephan, W. (2000). Joint effects of genetic hitchhiking and background selection on neutral variation. *Genetics*, 155(3):1415–1427.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*, 16(2):111–120.

Kingman, J. F. C. (1982). Tha coalescent. *Stoch. Proc. Applns.*, 13:235–238.

Kruglyak, L. (1999). Prospects for whole-genome linkage disequilibrium mapping of common disease genes. *Nat Genet*, 22(2):139–144.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., and co authors), e. . (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921.

Lercher, M. J., Smith, N. G. C., Eyre-Walker, A., and Hurst, L. D. (2002). The evolution of isochores: evidence from SNP frequency distributions. *Genetics*, 162(4):1805–1810.

Ling, B., Veazey, R. S., Luckay, A., Penedo, C., Xu, K., Lifson, J. D., and Marx, P. A. (2002). Siv(mac) pathogenesis in rhesus macaques of chinese and indian origin compared with primary hiv infections in humans. *AIDS*, 16(11):1489–1496.

Livingston, R. J., von Niederhausern, A., Jegga, A. G., Crawford, D. C., Carlson, C. S., Rieder, M. J., Gowrisankar, S., Aronow, B. J., Weiss, R. B., and Nickerson, D. A. (2004). Pattern of sequence variation across 213 environmental response genes. *Genome Res*, 14(10A):1821–1831.

Macaya, G., Thiery, J. P., and Bernardi, G. (1976). An approach to the organization of eukaryotic genomes at a macromolecular level. *J Mol Biol*, 108(1):237–254.

Marth, G. T., Czabarka, E., Murvai, J., and Sherry, S. T. (2004). The allele frequency spectrum in genome-wide human variation data reveals signals of differential demographic history in three large world populations. *Genetics*, 166(1):351–372.

McDonald, J. H. and Kreitman, M. (1991). Adaptive protein evolution at the Adh locus in Drosophila. *Nature*, 351(6328):652–654.

McVean, G. A. and Charlesworth, B. (2000). The effects of Hill-Robertson interference between weakly selected mutations on patterns of molecular evolution and variation. *Genetics*, 155(2):929–944.

Meunier, J. and Duret, L. (2004). Recombination drives the evolution of GC-content in the human genome. *Mol Biol Evol*, 21(6):984–990.

Morales, J. C. and Melnick, D. J. (1998). Phylogenetic relationships of the macaques (cercopithecidae: Macaca), as revealed by high resolution restriction site mapping of mitochondrial ribosomal genes. *J Hum Evol*, 34(1):1–23.

Mouchiroud, D., D'Onofrio, G., Assani, B., Macaya, G., Gautier, C., and Bernardi, G. (1991). The distribution of genes in the human genome. *Gene*, 100:181–187.

Nielsen, R. (2000). Estimation of population parameters and recombination rates from single nucleotide polymorphisms. *Genetics*, 154(2):931–942.

Nielsen, R. (2001). Statistical tests of selective neutrality in the age of genomics. *Heredity*, 86(Pt 6):641–647.

Nielsen, R., Bustamante, C., Clark, A. G., Glanowski, S., Sackton, T. B., Hubisz, M. J., Fledel-Alon, A., Tanenbaum, D. M., Civello, D., White, T. J., Sninsky, J. J., Adams, M. D., and Cargill, M. (2005a). A scan for positively selected genes in the genomes of humans and chimpanzees. *PLoS Biol*, 3(6):e170.

Nielsen, R., Hubisz, M. J., and Clark, A. G. (2004). Reconstituting the frequency spectrum of ascertained single-nucleotide polymorphism data. *Genetics*, 168(4):2373–2382.

Nielsen, R. and Wakeley, J. (2001). Distinguishing migration from isolation: a markov chain monte carlo approach. *Genetics*, 158(2):885–896.

Nielsen, R., Williamson, S., Kim, Y., Hubisz, M. J., Clark, A. G., and Bustamante, C. (2005b). Genomic scans for selective sweeps using SNP data. *Genome Res*, 15(11):1566–1575.

Peng, B. and Kimmel, M. (2005). simupop: a forward-time population genetics simulation environment. *Bioinformatics*, 21(18):3686–3687.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nat Genet*, 38(8):904–909.

Raveendran, M., Harris, R. A., Milosavljevic, A., Johnson, Z., Shelledy, W., Cameron, J., and Rogers, J. (2006). Designing new microsatellite markers for linkage and population genetic analyses in rhesus macaques and other nonhuman primates. *Genomics*, 88(6):706–710.

Rhesus Macaque Genome Sequencing and Analysis Consortium (2007). Evolutionary and biomedical insights from the rhesus macaque genome. *Science*, 316(5822):222–234.

Rogers, J., Garcia, R., Shelledy, W., Kaplan, J., Arya, A., Johnson, Z., Bergstrom, M., Novakowski, L., Nair, P., Vinson, A., Newman, D., Heckman, G., and Cameron, J. (2006). An initial genetic linkage map of the rhesus macaque (macaca mulatta) genome using human microsatellite loci. *Genomics*, 87(1):30–38.

Sanford, J., Baumgardner, J., Brewer, W., Gibson, P., and ReMine, W. (2007). Mendel's accountant: a biologically realistic forward-time population genetics program. *Scalable Computing: Practice and Experience*, 8:147–165.

Sawyer, S. A. and Hartl, D. L. (1992). Population genetics of polymorphism and divergence. *Genetics*, 132(4):1161–1176.

Siepel, A. and Haussler, D. (2004). Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Mol Biol Evol*, 21(3):468–488.

Slatkin, M. and Hudson, R. R. (1991). Pairwise comparisons of mitochondrial DNA sequences in stable and exponentially growing populations. *Genetics*, 129(2):555–562.

Sleator, D. D. and Tarjan, R. E. (1985). Self-adjusting binary search trees. *Journal of the ACM*, 32(3):652–686.

Smit, A. F. (1999). Interspersed repeats and other mementos of transposable elements in mammalian genomes. *Curr Opin Genet Dev*, 9(6):657–663.

Smith, D. G. and McDonough, J. (2005). Mitochondrial dna variation in chinese and indian rhesus macaques (macaca mulatta). *Am J Primatol*, 65(1):1–25.

Spencer, C. C. A. (2006). Human polymorphism around recombination hotspots. *Biochem Soc Trans*, 34(Pt 4):535–536.

Spencer, C. C. A., Deloukas, P., Hunt, S., Mullikin, J., Myers, S., Silverman, B., Donnelly, P., Bentley, D., and McVean, G. (2006). The influence of recombination on human genetic diversity. *PLoS Genet*, 2(9):e148.

Steiper, M. E. and Young, N. M. (2006). Primate molecular divergence dates. *Mol Phylogenet Evol*, 41(2):384–394.

Tajima, F. (1983). Evolutionary relationship of DNA sequences in finite populations. *Genetics*, 105(2):437–460.

Tajima, F. (1989). Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics*, 123(3):585–595.

Tajima, F. (1996). The amount of DNA polymorphism maintained in a finite population when the neutral mutation rate varies among sites. *Genetics*, 143(3):1457–1465.

Thiery, J. P., Macaya, G., and Bernardi, G. (1976). An analysis of eukaryotic genomes by density gradient centrifugation. *J Mol Biol*, 108(1):219–235.

Viray, J., Rolfs, B., and Smith, D. G. (2001). Comparison of the frequencies of major histocompatibility (mhc) class-ii dqa1 and dqb1 alleles in indian and chinese rhesus macaques (macaca mulatta). *Comp Med*, 51(6):555–561.

Wakeley, J. and Aliacar, N. (2001). Gene genealogies in a metapopulation. *Genetics*, 159(2):893–905.

Watterson, G. A. (1975). On the number of segregating sites in genetical models without recombination. *Theor Popul Biol*, 7(2):256–276.

Watterson, G. A. and Guess, H. A. (1977). Is the most frequent allele the oldest? *Theor Popul Biol*, 11(2):141–160.

Webster, M. T. and Smith, N. G. C. (2004). Fixation biases affecting human SNPs. *Trends Genet*, 20(3):122–126.

Webster, M. T., Smith, N. G. C., and Ellegren, H. (2003). Compositional evolution of noncoding DNA in the human and chimpanzee genomes. *Mol Biol Evol*, 20(2):278–286.

Weiss, R. A. (2001). Polio vaccines exonerated. *Nature*, 410(6832):1035–1036.

Williamson, S. and Orive, M. E. (2002). The genealogy of a sequence subject to purifying selection at multiple sites. *Mol Biol Evol*, 19(8):1376–1384.

Williamson, S. H., Hernandez, R., Fledel-Alon, A., Zhu, L., Nielsen, R., and Bustamante, C. D. (2005). Simultaneous inference of selection and population growth from patterns of variation in the human genome. *Proc Natl Acad Sci U S A*, 102(22):7882–7887.

Wiuf, C. (2006). Consistency of estimators of population scaled parameters using composite likelihood. *J Math Biol*, 53(5):821–841.

Yang, Z. (1996). Statistical properties of a DNA sample under the finite-sites model. *Genetics*, 144(4):1941–1950.

Zhang, Z. and Gerstein, M. (2003). Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res*, 31(18):5338–5348.